Communication-Free Multi-Agent Control Under Local Temporal Tasks and Relative-Distance Constraints

Meng Guo, *Student Member, IEEE*, Jana Tumova, *Member, IEEE*, and Dimos V. Dimarogonas, *Member, IEEE*

***Abstract*—We propose a distributed control and coordination strategy for multi-agent systems where each agent has a local task specified as a Linear Temporal Logic (LTL) formula and at the same time is subject to relative-distance constraints with its neighboring agents. The local tasks capture the temporal requirements on individual agents' behaviors, while the relative-distance constraints impose requirements on the collective motion of the whole team. The proposed solution relies only on relative-state measurements among the neighboring agents without the need for explicit information exchange. It is guaranteed that the local tasks given as syntactically co-safe or general LTL formulas are fulfilled and the relative-distance constraints are satisfied at all time. The approach is demonstrated with computer simulations.**

***Index Terms*—Agents and autonomous systems, cooperative control, hybrid systems, switched systems.**

I. Introduction

COOPERATIVE control of multi-agent systems generally focuses on designing local control laws to achieve a global control objective, such as reference-tracking [12], consensus [26], or formation [14]. In addition to these objectives, various relative-motion constraints are often imposed to ensure stability, safety and integrity of the overall system, such as collision avoidance [3], network connectivity [14], [31], or relative velocity constraints [12]. This work is motivated by the desire to specify and achieve more structured and complex team behaviors than the listed ones. Particularly, following a recent trend, we consider Linear Temporal Logic (LTL) formulas as suitable descriptions of desired high-level goals. LTL allows the designer to rigorously specify various temporal tasks, including periodic surveillance, sequencing, request-response, and their combinations. Furthermore, with the use of formal verification-inspired methods, a discrete plan that guarantees the specification satisfaction can be automatically synthesized, while various state space abstraction techniques bridge the continuous control problem and the discrete planning one. As a result, a generic hierarchical approach that allows for the correct-by-design control with respect to the given LTL specification has been formulated and largely employed during the last decade or so in single-agent as well as multi-agent settings. In particular, task specifications are expressed as LTL formulas for a single dynamical system in [4] and an automated framework is proposed to translate the task directly into a hybrid controller, which drives the system to fulfill this task. For multi-agent systems, LTL formulas have been used to specify complex high-level global tasks [2], [15], [17], [21], [22], [28], [30], local tasks [5], [9], [29] and even communication protocols [23] among the agents.

In temporal logic-based multi-agent control, two different points of view can be taken: a top-down and a bottom-up. In the former one, a global specification captures requirements on the overall team behavior. Typically, the focus of synthesizing a control strategy is on decomposing the global specification into smaller local tasks to be executed by the individual agents in a synchronized [2] or partially synchronized [17], [30] manner. A central monitoring unit is often crucial to ensure the satisfaction of the global goal.

In contrast, in the bottom-up approach, each agent is assigned its own local task. These tasks can be fully independent [5], [9] or partially dependent, involving requests for collaboration with the others [28], [29]. A major issue here is the decentralization of planning and control procedures. In [9], a decentralized revision scheme is suggested for a team in a partially-known workspace. In [5], gradual verification is employed to ensure that independent LTL tasks remain mutually satisfiable while avoiding collisions. In [29], a receding horizon approach is employed to achieve partially decentralized planning for collaborative tasks. In [28], the authors propose a compositional motion planning framework for multi-robot systems under local safe LTL specifications.

In this work, we tackle the multi-agent control problem under local LTL tasks from the bottom-up perspective. We are motivated by a scenario where the accomplishment of the local LTL tasks requires a collaborative execution of a sequence of services, which include 1) locally-assigned independent tasks that can be accomplished by a single robot, such as surveillance, delivery; 2) locally-assigned dependent tasks that require the involvement of other robots, such as collaborative assembly. Instead of challenging the centralized control ensuring the satisfaction of such dependent tasks, we view the problem as control under local LTL tasks and additional coupling constraints. Namely, the agents are subject to dynamic constraints with neighboring agents that maintain their geographical closeness necessary for the collaborative service execution. The

Manuscript received February 26, 2015; revised July 28, 2015 and January 19, 2016; accepted February 3, 2016. Date of publication February 11, 2016; date of current version December 2, 2016. This work is supported by EU STREP RECONFIG: FP7-ICT-2011-9-600825 and the Swedish Research Council (VR). Recommended by Associate Editor A. Girard.

The authors are with the ACCESS Linnaeus Center and Center for Autonomous Systems, KTH Royal Institute of Technology, SE-100 44, Stockholm, Sweden (e-mail: mengg@kth.se; tumova@kth.se; dimos@kth.se).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2016.2527731

maintenance of relative-distance constraints is closely related to the connectivity of the multi-agent network in robotic tasks [14]. As pointed out in [12], [31], [32], maintaining this connectivity is of great importance for the stability, safety and integrity of the overall team, for global objectives like rendezvous, formation and flocking. Very often the connectivity of underlying interaction graphs is imposed by assumption rather than treated as an extra control objective. We addressed a version of this problem in [10], where we proposed a dynamic leader-follower coordination and control scheme as a solution. In this work, however, we aim for a fully decentralized and communication-free solution that is applicable, e.g., to low-cost robotic systems equipped with range and angle sensors, but without communication capabilities.

The bottom-up viewpoint is especially fitting for heterogeneous systems, where certain tasks can be accomplished only by certain agents, and hence, it is reasonable to assume that each agent is given a task for which it is responsible. However, it is worth mentioning that global cooperative tasks can also be accommodated in our approach as long as they can be decomposed into local ones similarly as in [2], [30]. In particular, thanks to the geographical closeness due to the additional relative-distance constraints, this decomposition does not lead to locally-assigned independent tasks, nor to locally-assigned dependent tasks whose execution should be synchronized in time. In other words, the services over which the LTL formulas are built can be shared by several agents, but through the decomposition process they might be simply assigned to a single agent (owner of the respective task), instead of assigning it to the subset of involved agents that are later required to carefully synchronize to participate on providing them.

Our solution consists of four ingredients: an initial plan synthesis algorithm, a decentralized potential-field-based motion controller, a discrete switching policy and finally a plan adaptation algorithm. The application domains are multi-agent systems with fully-actuated agents within a 2D obstacle-free workspace where agents are assumed to be point mass and thus inter-agent collision is not considered. Tasks are assigned to the agents locally over (possibly collaborative) services provided in pre-defined regions of interests visited along the agent's trajectories. The main contributions are: (i) different from [5] and [10] where a satisfying discrete plan is enough, the proposed initial plan synthesis algorithm here minimizes a cost of a satisfying plan; (ii) the proposed distributed motion controller guarantees almost global convergence and the satisfaction of relative-distance constraints, for an *arbitrary* number of leaders with different local goals. Most related literature only allows for a single leader [10], [26] or multiple leaders with the same global goal as the team [24], [32]; and (iii) three different local coordination policies are proposed for different types of local tasks. Compared with [17], these coordination policies are fully-distributed and applied to communication-less agents. This paper builds on preliminary results from [11]. We provide, here, detailed proofs that are omitted, there. Moreover, we are interested to find the cost-optimal plan instead of just a plan as in [11]. We also enrich the analysis by considering mixed types of local task specifications. Additionally, a real-time plan adaptation algorithm is proposed.

The rest of the paper is organized as follows. Section II introduces the preliminaries. In Section III, we formalize the considered problem. Section IV presents our solution in details. Section V demonstrates the numerical simulations. We conclude and discuss about future directions in Section VI.

II. Preliminaries

A. Linear Temporal Logic (LTL)

A *Linear Temporal Logic (LTL)* formula over a set of *atomic propositions* Σ that can be evaluated as true or false is defined inductively according to the following rules [1]: an atomic proposition $\sigma \in \Sigma$ is an LTL formula; if φ and ψ are LTL formulas, then also $\neg\varphi$, $\varphi \wedge \psi$, $\bigcirc\varphi$, $\varphi \mathsf{U} \psi$, $\diamond\varphi$, and $\square\varphi$ are LTL formulas, where $\neg$ (*negation*), $\wedge$ (*conjunction*) are standard Boolean connectives and $\bigcirc$ (*next*), U (*until*), $\diamond$ (*eventually*), and $\square$ (*globally*) are temporal operators. The semantics of LTL is defined over the infinite words over 2^{Σ}. Intuitively, $\sigma \in \Sigma$ is satisfied on a word $w = w(1)w(2)\ldots$ if it holds at $w(1)$, i.e., if $\sigma \in w(1)$. Formula $\bigcirc\varphi$ holds true if φ is satisfied on the word suffix that begins in the next position $w(2)$, whereas $\varphi_1 \mathsf{U} \varphi_2$ states that φ_1 has to be true until φ_2 becomes true. Finally, $\diamond\varphi$ and $\square\varphi$ are true if φ holds on w eventually and always. For full details, see, e.g., [1].

Syntactically co-safe LTL (sc-LTL) is a subclass of LTL built without the operator $\square$ and with the restriction that the negation operator $\neg$ can be applied to atomic propositions only [19]. In contrast to general LTL formulas, the satisfaction of an sc-LTL formula can be achieved in a finite time, i.e., each word over 2^{Σ} satisfying an sc-LTL formula φ consists of a *satisfying prefix* that can be followed by an arbitrary suffix. The set of words that satisfies an LTL formula φ over Σ can alternatively be captured through a *Büchi automaton* $\mathcal{A}_{\varphi} = (Q, 2^{\Sigma}, \delta, Q_0, F)$, where Q is a finite set of states; 2^{Σ} is an input alphabet; $\delta : Q \times 2^{\Sigma} \to 2^{Q}$ is a transition relation; $Q_0 \subseteq Q$ is a set of initial states; and $F \subseteq Q$ is a set of accepting states. An infinite run over an input word $w = w(1)w(2)\ldots$ is an infinite sequence of states $\varrho = q_1 q_2 \ldots$, such that $q_1 \in Q_0$ and $q_{i+1} \in \delta(q_i, w(i))$, for all $i \geq 1$. A run is accepting if it intersects the set of accepting states F infinitely many times, and a word w is accepted if there exists an accepting run over it. Any LTL formula φ can be algorithmically translated [7] into a Büchi automaton.

B. Discrete Plan Synthesis

A weighted transition system is a tuple $\mathcal{T} = (S, \longrightarrow, s_0, W, \Sigma, L)$, where S is a finite set of states; $\longrightarrow \subseteq S \times S$ is a (deterministic) transition relation; $s_0 \in S$ is the initial state; $W : S \times S \to \mathbb{R}^{+}$ is a weight function; Σ is a set of atomic propositions; and $L : S \to 2^{\Sigma}$ is a labeling function. $W(s, s')$ approximates the cost of the transition $(s, s') \in \longrightarrow_i$. A finite or infinite run of $\mathcal{T}$ is a finite or infinite sequence of states $\tau = s_1 s_2 \ldots$, such that $s_1 = s_0$ is the initial state, and for all $i \geq 1$ it holds that $(s_i, s_{i+1}) \in \longrightarrow$. The *word* produced by the run $s_1 s_2 \ldots$ is $L(s_1)L(s_2)\ldots$.

The goal of *discrete plan synthesis* is to find a run of a given weighted transition system, such that φ is satisfied by the produced word. It can be addressed via the construction and analysis of a product automaton: a product automaton of a weighted transition system and a Büchi automaton is a tuple $\mathcal{P} = \mathcal{T} \otimes A_\varphi = (Q_\mathcal{P}, 2^\Sigma, \delta_\mathcal{P}, Q_{\mathcal{P},0}, F_{\mathcal{P},0}, W_\mathcal{P})$, where $Q_\mathcal{P} = S \times Q$; $\delta_\mathcal{P} \subseteq Q_\mathcal{P} \times 2^\Sigma \times Q_\mathcal{P}$. $((s,q), L(s), (s',q')) \in \delta_\mathcal{P}$ if $(s,s') \in \longrightarrow$ and $q' \in \delta(q, L(s))$; $Q_{\mathcal{P},0} = \{(s_0,q_0) | q_0 \in Q_0\}$; $F_{\mathcal{P},0} = \{(s,q_f) | s \in S, q_f \in F\}$; $W_\mathcal{P} : \delta_\mathcal{P} \to \mathbb{R}^+$. $W_\mathcal{P}((s,q), \sigma, (s',q')) = W(s,s')$, where $((s,q),\sigma,(s',q')) \in \delta_\mathcal{P}$. Each run of the product automaton $\varrho = (s_1,q_1)(s_2,q_2)\ldots$ can be projected onto a run $\tau = s_1 s_2 \ldots$ of $\mathcal{T}$ and onto a run $q_1, q_2 \ldots$ of $\mathcal{A}_\varphi$. Particularly, an accepting run of $\mathcal{P}$ projects onto a run of $\mathcal{T}$ that produces a word accepted by $\mathcal{A}_\varphi$. Hence, finding an accepting run of $\mathcal{P}$ gives a solution to the plan synthesis problem. The interested reader is referred, e.g., to [1] for further details.

C. Weighted Graph

An undirected weighted graph is a tuple $G = (\mathcal{N}, E, h)$, where $\mathcal{N} = \{1, \ldots, N\}$ is a set of nodes; $E \subseteq \mathcal{N} \times \mathcal{N}$ is a set of *edges*; and $h : E \to \mathbb{R}^+$ is the weight function, which can be omitted if the weight is uniform over all edges. Each node i has a set of *neighbors* $\mathcal{N}_i = \{j \in \mathcal{N} | (i,j) \in E\}$. A path from node i to j is a sequence of nodes starting with i and ending with j such that the consecutive nodes are neighbors. G is *connected* if there is a path between any two nodes and G is *complete* if $E = \mathcal{N} \times \mathcal{N}$. The Laplacian matrix $\mathbf{H}$ of G is an $N \times N$ positive semidefinite matrix: $\mathbf{H}(i,i) = \sum_{j \in \mathcal{N}_i} h(i,j), \forall\, i \in \mathcal{N}$; $\mathbf{H}(i,j) = -h(i,j), \forall\,(i,j) \in E$, and $\mathbf{H}(i,j) = 0$, otherwise. For a connected graph G, $\mathbf{H}$ has nonnegative eigenvalues [8] and a single zero eigenvalue with the eigenvector $\mathbf{1}_N$, where $\mathbf{1}_N = [1, \ldots, 1]^T$.

In this paper, each vector norm over $\mathbb{R}^n$ is the Euclidean norm [13]. We use $|S|$ to denote the cardinality of a set S and $v[i]$ to denote the i-th element of a vector or a sequence v.

III. Problem Formulation

A. Agent Dynamics and Network Structure

We consider a team of N autonomous agents with identities (IDs) $i \in \mathcal{N} = \{1, \ldots, N\}$, satisfying the dynamics:

$$\dot{x}_i(t) \triangleq u_i(t), \qquad i \in \mathcal{N} \tag{1}$$

where $x_i(t), u_i(t) \in \mathbb{R}^2$ are the respective state and the control input of agent i at time $t > 0$. Let $x_i(0)$ be the given initial state. The agents are modeled as point masses without volume, i.e., inter-agent collisions are not considered.

Each agent has a sensing radius $r > 0$, which is assumed to be identical for all agents. Namely, each agent can only observe another agent' state if their relative distance is less than r. Thus, given $\{x_i(0), i \in \mathcal{N}\}$, we define the embedded graph $G_0(t) \triangleq (\mathcal{N}, E_0(t))$, where $(i,j) \in E_0(t)$ if $\|x_i(t) - x_j(t)\| < r$. We assume that $G_0(0)$ is connected initially and one of the control objectives is to ensure that $G_0(t)$ remains connected for all time $t \geq 0$.

B. Task Specifications

Within the 2D workspace, each agent $i \in \mathcal{N}$ has a set of $M_i \geq 1$ regions of interest, denoted by $\Pi_i \triangleq \{\pi_{i1}, \ldots, \pi_{iM_i}\}$. These regions can be of different shapes, such as spheres, triangles, or polygons. For simplicity of presentation, $\pi_{i\ell} \in \Pi_i$ is here represented by a circular area around a point of interest:

$$\pi_{i\ell} = \mathcal{B}(c_{i\ell}, r_{i\ell}) = \{y \in \mathbb{R}^2 \mid \|y - c_{i\ell}\| \leq r_{i\ell}\} \tag{2}$$

where $c_{i\ell} \in \mathbb{R}^2$ is the center; $r_{\min} \leq r_{i\ell} \leq r_{\max}$ is the radius where $0 < r_{\min} < r_{\max}$ are the upper and lower bounds for the radii for all regions. Other shapes than spheres would require an under-approximation of these shapes as spheres first, in order to apply the proposed solution. We assume that these regions do not intersect and the workspace is bounded, particularly:

Assumption 1: (I) $\|c_{i\ell}\| < c_{\max}, \forall\, i \in \mathcal{N}$ and $\forall\, \pi_{i\ell} \in \Pi_i$, where $c_{\max} > 0$ is a given constant. (II) $\|c_{i\ell_i} - c_{j\ell_j}\| > 2r_{\max}, \forall\, i,j \in \mathcal{N}, \forall\, \pi_{i\ell_i} \in \Pi_i$ and $\forall\, \pi_{j\ell_j} \in \Pi_j$.

Moreover, there is a set of atomic propositions known to agent i, denoted by Σ_i. Each region of interest is associated with a subset of Σ_i through the labeling function $L_i : \Pi_i \to 2^{\Sigma_i}$. Without loss of generality, we assume that $\Sigma_i \cap \Sigma_j = \emptyset$, for all $i, j \in \mathcal{N}$ such that $i \neq j$. We view the atomic propositions $L_i(\pi_{i\ell})$ as a set of *services* that agent i can provide when being present in region $\pi_{i\ell} \in \Pi_i$. Hence, upon the visit to $\pi_{i\ell}$, the agent i chooses among $L_i(\pi_{i\ell})$ the subset of atomic propositions to be evaluated as true, i.e., the subset of services it *provides* among the available ones. These services are abstractions of action primitives that can be executed in different regions, such as manipulation tasks or data gathering. Some services within Σ_i may depend on the other agents' collaborations, meaning that they can be provided only if the other agents are around. In this paper, we do not focus on how services are being provided or how the agents collaborate in providing them; we aim at guaranteeing the necessary preconditions for providing these collaborative services: the geographical closeness.

We denote by $\mathbf{x}_i(T)$ the *trajectory* of agent i during the time interval $[0, T)$, where $T > 0$ and T can be infinity. The trajectory $\mathbf{x}_i(T)$ is associated with a unique finite or infinite sequence, called a *path*, $\mathbf{p}_i(T) \triangleq \pi_{i1}\pi_{i2}\ldots$ of regions in Π_i that agent i crosses, and with a finite or infinite sequence of time instants $t'_{i0} t_{i1} t'_{i1} t_{i2} t'_{i2} \cdots$ when i enters or leaves the respective regions. Formally, for all $k \geq 1$: $0 = t'_{i0} \leq t_{ik} \leq t'_{ik} < t_{ik+1} < T$, $x_i(t) \in \pi_{ik}$, for $\pi_{ik} \in \Pi_i$, $\forall\, t \in [t_{ik}, t'_{ik}]$, and $x_i(t) \notin \pi_{i\ell}$, $\forall\, \pi_{i\ell} \in \Pi_i$ and $\forall\, t \in (t'_{ik-1}, t_{ik})$. However, agent i may choose to provide services only at some regions along the path $\mathbf{p}_i$. Denote by $\overline{\mathbf{p}}_i(T) = \pi_{i\ell_1}\pi_{i\ell_2}\ldots$ the *effective path* as a subsequence of $\mathbf{p}_i$ such that $\ell_k < \ell_{k+1}, \forall\, k \geq 1$ and $\pi_{i\ell_k} \in \mathbf{p}_i(T)$, $\forall\, \pi_{i\ell_k} \in \overline{\mathbf{p}}_i(T)$. The *word* produced by agent i is given by the provided services along the sequence of regions in $\overline{\mathbf{p}}_i$. In particular, at region for $\pi_{i\ell_k} \in \overline{\mathbf{p}}_i(T)$, agent i chooses to provide a set of services w_{ℓ_k}, where $w_{\ell_k} \neq \emptyset$ and $w_{\ell_k} \subseteq L_i(\pi_{i\ell_k})$ is a subset of services available at region $\pi_{i\ell_k}$. Namely, the produced word $\texttt{word}_i(T) = w_{\ell_1} w_{\ell_2} \cdots$ complies with $\overline{\mathbf{p}}_i(T)$ if $\emptyset \subset w_{\ell_k} \subseteq L_i(\pi_{i\ell_k})$, $\forall\, \pi_{i\ell_k} \in \overline{\mathbf{p}}_i(T)$. Thus, an agent's behavior is fully determined by its trajectory, its effective path and the word it produces.

The local task of each agent $i \in \mathcal{N}$ is specified as a general LTL or an sc-LTL formula φ_i over Σ_i and captures requirements on the services to be provided by agent i. In this work, we do not focus on how the service providing is executed by an agent; we only aim at controlling an agent's motion to reach the regions where these services are available. Given the trajectory $\mathbf{x}_i(T)$ of agent i, the satisfaction of its task formula φ_i is defined as follows:

Definition 1: Agent i's trajectory $\mathbf{x}_i(T)$ satisfies φ_i if there exists an effective path $\overline{\mathbf{p}}_i(T)$ and a compliant word $\texttt{word}_i(T)$ such that $\texttt{word}_i(T) \models \varphi_i$.

C. Cost of an Effective Path

Since we are interested in the quantitative cost of satisfying a local task, we propose the following way to measure the cost of an effective path. The motion of agent i is estimated through a weighted transition system [1]:

$$\mathcal{T}_i \triangleq \left(\Pi_i', \longrightarrow_i, L_i, \Sigma_i, \pi_{i,0}', W_i\right) \tag{3}$$

where $\Pi_i' \triangleq \Pi_i \cup \{\pi_{i0}\}$. $\pi_{i0} \triangleq x_i(0)$ represents the agent's initial position symbolically; $\longrightarrow_i \triangleq \Pi_i' \times \Pi_i'$ is the transition relation, which is the full Cartesian product; L_i and Σ_i are the labelling function and the set of propositions for services defined earlier; $\pi_{i,0}' \triangleq \pi_{i0}$ is the initial state; $W_i : \longrightarrow_i \rightarrow \mathbb{R}^+$ approximates the cost of each transition, $W_i(\pi_{i\ell_1}, \pi_{i\ell_2}) \triangleq \|c_{i\ell_1} - c_{i\ell_2}\| - r_{i\ell_1} - r_{i\ell_2}$, $\forall (\pi_{i\ell_1}, \pi_{i\ell_2}) \in \longrightarrow_i$, where $\pi_{i\ell_1}, \pi_{i\ell_2} \in \Pi_i$; and $W_i(\pi_{i0}, \pi_{i\ell}) \triangleq \|x_i(0) - c_{i\ell}\|$.

Consider that one of agent i's effective path is given by $\overline{\mathbf{p}}_i(T) = \pi_{i\ell_1}\pi_{i\ell_2}\ldots$. Then this effective path is assigned with a cost. In this work, the cost is defined as the maximal distance traveled between two consecutive regions along the path. Formally, denote by ϑ the set of consecutive regions in $\overline{\mathbf{p}}_i(T)$, i.e., $\vartheta = \{(\pi_{i0}', \pi_{\ell_1}), (\pi_{\ell_k}, \pi_{\ell_{k+1}}), \forall\, k = 1, 2, \ldots\}$. Its cost is defined as:

$$\boldsymbol{cost}\left(\overline{\mathbf{p}}_i(T)\right) \triangleq \max_{(\pi_s, \pi_g) \in \vartheta} \{W_i(\pi_s, \pi_g)\} \tag{4}$$

which is still valid when $T = \infty$. Namely, we want to minimize the maximal distance travelled between two consecutive regions in the effective path. This is due to the consideration that for services-related specifications it is of great interest to ensure the frequency at which a service is provided. The standard cost definition as the cumulative cost can be used instead by incorporating the synthesis algorithms proposed in [9], [15]. Formally, the problem we consider is stated below:

Problem 1: Given a team of N agents as in Section III-A, and their task specifications as in Section III-B, design a distributed control law u_i, the associated effective path $\overline{\mathbf{p}}_i(T)$ and its corresponding word $\texttt{word}_i(T), \forall\, i \in \mathcal{N}$, such that for $T = \infty$:

1) $\texttt{word}_i(T)$ satisfies φ_i; and
2) $\overline{\mathbf{p}}_i(T)$ has minimal cost by (4); and
3) $\|x_i(t) - x_j(t)\| < r, \forall\, (i,j) \in E_0(0), \forall\, t \in [0, T)$.

Note that the task specifications $\varphi_i, \forall\, i \in \mathcal{N}$ need not be the same type among the agents, i.e., φ_i can be either a syntactically co-safe or a general LTL formula. More details can be found in Section IV-C. Moreover, two practical applications of the above formulation are that (i) a team of underwater vehicles are deployed such that each vehicle monitors a certain feature (e.g., temperature, concentration) over different areas. Local task specifications are preferred over a global one as the vehicles have distinctive capabilities and clear task assignments. Since explicit wireless communication is unreliable in underwater domain, sonar-based range sensors are more practical for our communication-free solution, while the relative-distance constraint can be formulated as the formation requirement of the team for certain collaborative tasks; (ii) different users specify a local task to different robots within a heterogeneous team. Without any wireless communication, one robot needs to transfer its knowledge about the workspace by gestures to other teammates, which requires that the robots stay close to enable the gesture detection, e.g., see [18].

IV. Solution

The proposed solution consists of four layers: (i) an offline synthesis scheme for the initial discrete plan of each agent, i.e., the effective path of progressive goal regions and the sequence of services to be provided; (ii) a distributed continuous control scheme that guarantees that one of the agents reaches its progressive goal region in finite time while the relative-distance constraints are fulfilled at all time; (iii) a hybrid control layer that coordinates the discrete plan execution and the continuous control law switching, to ensure the satisfaction of each agent's local task; and (iv) a real-time plan adaptation algorithm that each agent could apply to improve its discrete plan, given its updated position and its plan execution status.

A. Initial Discrete Plan Synthesis

As introduced in Section II, the discrete plan satisfying the local task can be generated using standard techniques from automata-based formal synthesis. Loosely speaking, an LTL or an sc-LTL formula φ_i is firstly translated into a Büchi automaton. A product automaton is constructed as described in Section II-B. Then, one of its accepting runs is found and projected onto the desired discrete plan.

Here, we aim to find an effective path for agent $i \in \mathcal{N}$ with the property that (i) there exists a compliant word satisfying φ_i and (ii) the effective path is optimal with respect to the cost function (4). To that end, we adapt the standard synthesis scheme as follows. For each agent $i \in \mathcal{N}$, we build a product automaton $\mathcal{P}_i = \mathcal{T}_i \otimes \mathcal{A}_{\varphi_i} = (Q_{\mathcal{P},i}, 2^{\Sigma_i}, \delta_{\mathcal{P},i}, Q_{\mathcal{P},i,0}, F_{\mathcal{P},i}, W_{\mathcal{P},i})$ as described in Section II-B, with a slight change in $\delta_{\mathcal{P},i}$ reflecting the form of the words compliant with the effective paths: $((s,q), \sigma, (s',q')) \in \delta_{\mathcal{P}}$ if $(s,s') \in \longrightarrow$ and $q' \in \delta(q,\sigma)$, where $\emptyset \subset \sigma \subseteq L(s)$. An accepting run ϱ_i of $\mathcal{P}_i$ over an input word w_i projects onto a run τ_i of $\mathcal{T}_i$ with the property that w complies with the effective path $\overline{\mathbf{p}}_i(T)$ and satisfies φ_i, while for any word w_i that satisfies φ_i and is compliant with an effective path $\overline{\mathbf{p}}_i(T)$, there exists an accepting run of $\mathcal{P}_i$ that projects onto a run τ_i of $\mathcal{T}_i$ that is compliant with w.

Algorithm 1: Minimum bottleneck path, $\texttt{MinBot}(v)$

Input: Product automaton $\mathcal{P}_i$ and $v \in Q_{\mathcal{P},i}$
Output: $\texttt{D}_v, \texttt{P}_v$
$S := Q_{\mathcal{P},i}$; $\texttt{D}[v] := 0$;
forall the $u \in Q_{\mathcal{P},i}$ **do**
  if $u \neq v$ **then**
    $\texttt{D}_v(u) := \infty$; $\texttt{P}_v(u) := None$;
while $S \neq \emptyset$ **do**
  $u := \text{argmin}_{u\in S}\{\texttt{D}_v(u)\}$; remove u from S;
  forall the $u' \in \delta_{\mathcal{P},i}(u)$ **do**
    $b := \max\{\texttt{D}_v(u), W_{\mathcal{P},i}(u,u')\}$;
    if $b \leq \texttt{D}_v(u')$ **then**
      $\texttt{D}_v(u') := b$; $\texttt{P}_v(u') := u$;
return $\texttt{D}_v, \texttt{P}_v$

In Algorithm 1, we modify the Dijkstra's algorithm (see, e.g., [20]) to find the finite paths from a state $v \in Q_{\mathcal{P},i}$ to all the other states minimizing the bottleneck weight, where the bottleneck weight of a path is defined as the maximal weight of the individual edges on the path. The output of the $\texttt{MinBot}(v)$ algorithm is the distance function $\texttt{D}_v : Q_{\mathcal{P},i} \to \mathbb{R}^+$, where $\texttt{D}_v(u)$ gives the minimal bottleneck weight from v to the state $u \in Q_{\mathcal{P},i}$, and the predecessor function $\texttt{P}_v : Q_{\mathcal{P},i} \to Q_{\mathcal{P},i}$, where $\texttt{P}_v(u)$ gives the predecessor of $u \in Q_{\mathcal{P},i}$ on the minimal-bottleneck path.

Then, we can synthesize the optimal effective path of $\mathcal{P}_i$ as follows:

(I) For all $v_0 \in Q_{\mathcal{P},i,0}$, compute $(\texttt{D}_{v_0}, \texttt{P}_{v_0}) = \texttt{MinBot}(v_0)$. (II) For all $v_f \in F_{\mathcal{P},i}$, compute $(\texttt{D}_{v_f}, \texttt{P}_{v_f}) = \texttt{MinBot}(v_f)$. (III) Find the pair (v_0, v_f), where $v_0 \in Q_{\mathcal{P},i,0}$ and $v_f \in F_{\mathcal{P},i}$ that minimizes the term $\max\{\texttt{D}_{v_0}(v_f), \max_{(v,v_f)\in\delta_{\mathcal{P},i}}\{\texttt{D}_{v_f}(v), W_{\mathcal{P},i}(v,v_f)\}\}$, where $\texttt{D}_{v_0}(v_f)$ is the minimal bottleneck from v_0 to v_f; and the second term is the minimal bottleneck from v_f back to itself. Note that v is any predecessor of v_f given by $\delta_{\mathcal{P},i}$. Denote the optimal pair as $(v_0^\star, v_f^\star)$. (IV) The computed accepting run of $\mathcal{P}_i$ is in the prefix-suffix form $\varrho_i = \varrho_{i,\text{pre}}(\varrho_{i,\text{suf}})^\omega$, where $\varrho_{i,\text{pre}}$ is the minimal-bottleneck path from $v_0^\star$ to $v_f^\star$, computed based on $P_{v_0^\star}$; $\varrho_{i,\text{suf}}$ is the minimal-bottleneck cycle from $v_f^\star$ back to itself, computed based on $P_{v_f^\star}$ similarly.

The accepting run ϱ_i of $\mathcal{P}_i$ is naturally projected onto $\mathcal{T}_i$ as follows, which results into the *initial plan*: $\tau_i(0) = \tau_{i,\text{pre}}(\tau_{i,\text{suf}})^\omega$, where $\tau_{i,\text{pre}} = (\pi_{i1}, w_{i1}) \cdots (\pi_{ik_i}, w_{ik_i})$ is the *plan prefix*, and $\tau_{i,\text{suf}} = (\pi_{ik_i+1}, w_{ik_i+1}) \cdots (\pi_{iK_i}, w_{iK_i})$ is the periodical *plan suffix*; (π_{ik}, w_{ik}) is called the *progressive goal region*; $\pi_{ik} \in \Pi_i$ and $\emptyset \subset w_{ik} \subseteq L_i(\pi_{ik})$, $\forall\, k = 1, \ldots, K_i$. Thus, the word corresponding to $\tau_i(0)$ is given by its projection onto Σ_i, namely $\texttt{word}_i(T) = \tau_i(0)|_{\Sigma_i} = w_{i1} \cdots w_{ik_i}(w_{ik_i+1} \cdots w_{iK_i})^\omega$; then the effective path $\overline{\mathbf{p}}_i$ is given as the projection of $\tau_i(0)$ onto Π_i, namely, $\overline{\mathbf{p}}_i(T) = \tau_i(0)|_{\Pi_i} = \pi_{i1} \cdots \pi_{ik_i}(\pi_{ik_i+1} \cdots \pi_{iK_i})^\omega$. We denote by $\overline{\mathbf{p}}_{i,\text{pre}}(T) = \pi_{i1} \cdots \pi_{ik_i}$ the prefix of the effective path and by $\overline{\mathbf{p}}_{i,\text{suf}}(T) = \pi_{ik_i+1} \cdots \pi_{iK_i}$ the suffix. For general LTL formulas φ_i, the plan τ_i indicates the desired effective path $\overline{\mathbf{p}}_i(T)$ and the infinite sequence of services $\texttt{word}_i(T)$; for sc-LTL formulas φ_i, $\tau_{i,\text{pre}}$ indicates the desired effective path $\overline{\mathbf{p}}_i(T)$ and the finite sequence of services $\texttt{word}_i(T)$. Moreover, any trajectory that satisfies the prefix with an arbitrary extension would satisfy φ_i. Note that, in our work, LTL formulas are interpreted over the *provided* services along a trajectory, not the available ones. Hence, unplanned crossing of a region as well as collaborations on providing other agents' services does not influence the local LTL task satisfaction.

The computational complexity of the above synthesis algorithm is $\mathcal{O}(|Q_{\mathcal{P},i}|^2 \cdot |Q_{\mathcal{P},i,0}| \cdot |F_{\mathcal{P},i}|)$ in the worst case, where $|Q_{\mathcal{P},i}|$ is the number of states in $\mathcal{P}_i$. Other recent temporal logic-based discrete plan synthesis algorithms can be used to accommodate various environmental constraints and advanced plan optimality criteria, e.g., [1], [9], [30].

B. Continuous Controller Design

As stated previously, each agent synthesizes its initial plan as a sequence of goal regions to reach and a set of services to provide there. However, these goal regions of different agents can be at different locations and potentially far away. Furthermore, the relative-distance constraints require the neighboring agents to stay close. Before stating the control scheme, let us first introduce the notion of connectivity graph, which allows us to handle the relative-distance constraints. Recall that each agent has a limited sensing radius $r > 0$ as mentioned in Section III-A. Let $\kappa \in (0, r)$ be a given constant. Then, we define the connectivity graph $G(t)$ as follows:

Definition 2: Let $G(t) \triangleq (\mathcal{N}, E(t))$ denote the undirected time-varying connectivity graph at time $t \geq 0$, where $E(t) \subseteq \mathcal{N} \times \mathcal{N}$ is the set of edges. (I) $G(0) = G_0(0)$; (II) At time $t > 0$, $(i,j) \in E(t)$ iff one of the following conditions hold: (i) $\|x_i(t) - x_j(t)\| \leq r - \kappa$; or (ii) $r - \kappa < \|x_i(t) - x_j(t)\| \leq r$ and $(i,j) \in E(t^-)$, where $t^- < t$ and $|t - t^-| \to 0$.

Note that the condition (II) above guarantees that a new edge will only be added when the distance between two previously-unconnected agents decreases below $r - \kappa$. In other words, there is a hysteresis effect when adding new edges to the connectivity graph. Consequently, each agent $i \in \mathcal{N}$ has a time-varying set of neighbors $\mathcal{N}_i(t) = \{j \in \mathcal{N} | (i,j) \in E(t)\}$. Let the progressive goal region of agent $i \in \mathcal{N}$ at time t be given by $\pi_{\text{ig}} = \mathcal{B}(c_{\text{ig}}, r_{\text{ig}}) \in \Pi_i$. We propose the following two different control modes:

1) the *active* mode:

$$\mathbf{C}_{\text{act}}: \quad u_i(t) \triangleq -d_i p_i - \sum_{j\in\mathcal{N}_i(t)} h_{\text{ij}} x_{\text{ij}} \tag{5}$$

2) the *passive* mode:

$$\mathbf{C}_{\text{pas}}: \quad u_i(t) \triangleq -\sum_{j\in\mathcal{N}_i(t)} h_{\text{ij}} x_{\text{ij}} \tag{6}$$

where $x_{\text{ij}} \triangleq x_i - x_j$; $p_i \triangleq x_i - c_{\text{ig}}$; and the coefficients are

$$d_i \triangleq \frac{\varepsilon^3}{(\|p_i\|^2 + \varepsilon)^2} + \frac{\varepsilon^2}{2\,(\|p_i\|^2 + \varepsilon)}; \; h_{\text{ij}} \triangleq \frac{r^2}{(r^2 - \|x_{\text{ij}}\|^2)^2}$$

where $\varepsilon > 0$ is a *key* design parameter to be appropriately tuned. We show in detail how to choose ε in the sequel. Note that both controllers in (5) and (6) are nonlinear and rely on only locally-available states, i.e., $x_i(t)$ and $x_j(t)$, $j \in \mathcal{N}_i(t)$.

Assume that $G(T_s)$ is connected at time $T_s > 0$. Moreover, assume that there are $1 \leq N_a \leq N$ agents that are in the *active* mode obeying (5) with its goal region as $\pi_{\rm ig} = \mathcal{B}(c_{\rm ig}, r_{\rm ig}) \in \Pi_i$; and the rest $N_p = N - N_a$ agents that are in the *passive* mode obeying (6). For simplicity, denote by the group of active and passive agents $\mathcal{N}_a, \mathcal{N}_p \subseteq \mathcal{N}$, respectively. In the rest of this section, we show that under *arbitrary* number of active agents, by following the control laws (5) and (6), exactly *one* active agent can reach its goal region within finite time $T_f \in (T_s, +\infty)$, while the relative distance $\|x_i(t) - x_j(t)\| < r, \forall\,(i,j) \in E(T_s)$ and $\forall\, t \in [T_s, T_f]$.

1) Relative-Distance Maintenance: In this part, we show that the relative-distance constraints are always satisfied under the control laws (5) and (6). We consider the potential-field function below:

$$V(t) \triangleq \frac{1}{2}\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}_i(t)} \phi_c(x_{\rm ij}) + b_i \sum_{i\in\mathcal{N}} \phi_g(x_i) \tag{7}$$

where $\phi_c(\cdot)$ is an attractive potential to agent i's neighbors:

$$\phi_c(x_{\rm ij}) \triangleq \frac{1}{2}\frac{\|x_{\rm ij}\|^2}{r^2 - \|x_{\rm ij}\|^2}, \qquad \|x_{\rm ij}\| \in [0, r-\delta) \tag{8}$$

while $\phi_g(\cdot)$ is an attractive force to agent i's goal, defined by:

$$\phi_g(x_i) \triangleq \frac{\varepsilon^2}{2}\frac{\|p_i\|^2}{\|p_i\|^2 + \varepsilon} + \frac{\varepsilon^2}{4}\ln\left(\|p_i\|^2 + \varepsilon\right) \tag{9}$$

where function $\ln(\cdot)$ is the natural logarithm; $b_i \in \mathbb{B}$ indicates the agent i's control mode. Namely, $b_i = 1, \forall\, i \in \mathcal{N}_a$ and $b_i = 0, \forall\, i \in \mathcal{N}_p$. Clearly $V(t)$ is lower-bounded by $N_a \ln(\varepsilon)\varepsilon^2/4$. Moreover, it can be verified that

$$\nabla_{x_i} V = \frac{\partial V}{\partial x_i} = b_i d_i p_i + \sum_{j\in\mathcal{N}_i(t)} h_{\rm ij} x_{\rm ij} = -u_i. \tag{10}$$

Theorem 1: $G(t)$ remains connected and no existing edges within $E(T_s)$ will be lost, namely, $E(T_s) \subseteq E(t), \forall\, t \geq T_s$.

Proof: Assume that the network $G(t)$ remains *invariant* during the time period $[t_1, t_2) \subseteq [T_s, \infty)$, i.e., no edges are added or removed. Thus the neighboring sets $\{\mathcal{N}_i, i \in \mathcal{N}\}$ also remain invariant and $V(t)$ is differentiable for $t \in [t_1, t_2)$. Then the time derivative of $V(t)$ is given by

$$\begin{aligned}\dot{V}(t) &= \sum_{i\in\mathcal{N}} (\nabla_{x_i} V)^T u_i \\ &= -\sum_{i\in\mathcal{N}} \left\| b_i d_i p_i + \sum_{j\in\mathcal{N}_i(t)} h_{\rm ij} x_{\rm ij} \right\|^2 \leq 0\end{aligned} \tag{11}$$

meaning that $V(t)$ is non-increasing, $\forall\, t \in [t_1, t_2)$. Thus, $V(t) \leq V(T_s) < +\infty$ for $t \in [t_1, t_2)$. On the other hand, assume a *new* edge (p, q) is added to $G(t)$ at $t = t_2$, where $p, q \in \mathcal{N}$. By Definition 2, $\|x_{pq}(t_2)\| \leq r - \kappa$ and $\phi_c(x_{pq}(t_2)) = ((r-\kappa)^2/\kappa(2r-\kappa)) < +\infty$ since $0 < \kappa < r$. Denote by $\widehat{E} \subset \mathcal{N} \times \mathcal{N}$ the set of newly-added edges at $t = t_2$. Let $V(t_2^+)$ and $V(t_2^-)$ be the value of $V(t)$ before and after adding the set of new edges to $G(t)$ at $t = t_2$. We get $V(t_2^+) = V(t_2^-) + \sum_{(p,q)\in\widehat{E}} \phi_c(x_{pq}(t_2)) \leq V(t_2^-) + |\widehat{E}|((r-\kappa)^2/\kappa(2r-\kappa)) < +\infty$, where we use the fact that $|\widehat{E}|$ is bounded as $\widehat{E} \subset \mathcal{N} \times \mathcal{N}$. Thus, $V(t) < +\infty$ also holds when new edges are added at time t_2. Similar analysis can be found in [14]. As a result, $V(t) < +\infty$ for $t \in [T_s, \infty)$. Note that $V(t)$ is non-increasing when $G(t)$ remains unchanged and increases when new edges are added. By Definition 2, one existing edge $(i, j) \in E(t)$ will be lost only if $x_{\rm ij}(t) = r$. It implies that $\phi_c(x_{\rm ij}) \to +\infty$ and $V(t) \to +\infty$ by (7). By contradiction, new edges might be added into $E(t)$, but no existing edges within $E(t)$ will be lost, namely, $E(T_s) \subseteq E(t), \forall\, t \geq T_s$. If $G(T_s)$ is initially connected, then $G(t)$ remains connected for all $t \geq T_s$. □

2) Convergence Analysis: In this part, we analyze in detail the convergence properties of the closed-loop system, i.e., the multi-agent system under the control laws (5) and (6) for *any* number of active and passive agents. We have shown that the potential function $V(t)$ is lower-bounded and non-increasing when $G(t)$ remains invariant by Theorem 1 above. We first show that the graph $G(t)$ becomes complete and thus invariant when the system converges to the set of critical points defined in the sequel. By LaSalle's invariance principle [16] we only need to find out the largest invariant set within $\{x_i, \forall\, i \in \mathcal{N} | \dot{V}(t) = 0\}$. By enforcing $\dot{V}(t) = 0$, it implies:

$$b_i d_i p_i + \sum_{j\in\mathcal{N}_i(t)} h_{\rm ij} x_{\rm ij} = 0, \quad \forall\, i \in \mathcal{N}. \tag{12}$$

Then, we can construct one $N \times N$ diagonal matrix $\mathbf{D}$ that $\mathbf{D}(i,i) = b_i d_i, \forall\, i \in \mathcal{N}$ and $\mathbf{D}(i,j) = 0, i \neq j$ and $i, j \in \mathcal{N}$. And, another $N \times N$ matrix $\mathbf{H}$ that $\mathbf{H}(i,i) = \sum_{j\in\mathcal{N}_i} h_{\rm ij}$, $\forall\, i \in \mathcal{N}$ and $\mathbf{H}(i,j) = -h_{\rm ij}$, $i \neq j$ and $\forall\,(i,j) \in E(t)$ while $\mathbf{H}(i,j) = 0, \forall\,(i,j) \notin E(t)$. Note that $h_{\rm ij} > 0$ as $\|x_{\rm ij}\| \in [0, r)$ by (10), $\forall\,(i,j) \in E(t)$. As a result, $\mathbf{H}$ is the Laplacian matrix of the graph $G(t) = (\mathcal{N}, E(t), h)$, where $h(i,j) = h_{\rm ij}$, $\forall\,(i,j) \in E(t)$. Then, (12) is equivalent to:

$$\mathbf{H} \otimes \mathbf{I}_2 \cdot \mathbf{x} + \mathbf{D} \otimes \mathbf{I}_2 \cdot (\mathbf{x} - \mathbf{c}) = 0 \tag{13}$$

where $\otimes$ is the Kronecker product [13]; $\mathbf{x}$ is the stack vector for x_i, $i \in \mathcal{N}$ and $\mathbf{x}[i] = x_i$; $\mathbf{I}_2$ is the 2×2 identity matrix; $\mathbf{c}$ is the stack vector for $c_{\rm ig}$ and $\mathbf{c}[i] = c_{\rm ig}$ if $i \in \mathcal{N}_a$ and $\mathbf{c}[i] = \mathbf{0}_2$ if $i \in \mathcal{N}_p$, where $\mathbf{0}_2$ is a 2×1 zero vector. Let $\mathcal{C}$ be the set of critical points of $V(t)$ that satisfy (13), i.e., $\mathcal{C} \triangleq \{x \in \mathbb{R}^{2N} | \mathbf{H} \otimes \mathbf{I}_2 \cdot \mathbf{x} + \mathbf{D} \otimes \mathbf{I}_2 \cdot (\mathbf{x} - \mathbf{c}) = 0\}$. Now, we show that at the critical points within $\mathcal{C}$ the relative distances between any two agents can be made arbitrarily small by reducing ε and, thus, the underlying network becomes a complete graph.

Lemma 2: For all critical points $\mathbf{x}_c \in \mathcal{C}$, (I) $\|x_{\rm ij}\|$ can be made arbitrarily small by reducing $\varepsilon, \forall\,(i,j) \in E(t)$; and (II) there exists $\varepsilon_0 > 0$ that if $\varepsilon < \varepsilon_0$, then the connectivity graph $G(t)$ is complete.

Proof: (I) For a critical point $\mathbf{x}_c \in \mathcal{C}$, $\sum_{(i,j)\in E(t)} h_{\rm ij}\|x_{\rm ij}\|^2 = \mathbf{x}_c^T \cdot (\mathbf{H} \otimes \mathbf{I}_2) \cdot \mathbf{x}_c$. holds. Combining it with (13), we get

$$\begin{aligned}\sum_{(i,j)\in E(t)} & h_{\rm ij}\|x_{\rm ij}\|^2 \\ &= -\mathbf{x}_c^T \cdot (\mathbf{D} \otimes \mathbf{I}_2) \cdot (\mathbf{x}_c - \mathbf{c}) \\ &= -(\mathbf{x}_c - \mathbf{c})^T \cdot (\mathbf{D} \otimes \mathbf{I}_2) \cdot (\mathbf{x}_c - \mathbf{c}) - \mathbf{c}^T \cdot (\mathbf{D} \otimes \mathbf{I}_2) \\ &\quad \cdot (\mathbf{x}_c - \mathbf{c}) \\ &= -\sum_{i\in\mathcal{N}} b_i d_i \left(\|p_i\|^2 + c_{i\ell}^T p_i\right) \leq \sum_{i\in\mathcal{N}} b_i \|c_{i\ell}\| d_i \|p_i\|.\end{aligned}$$

Since it can be verified that $d_i\|p_i\| < \varepsilon\sqrt{\varepsilon}$ for $\|p_i\| \geq 0$ and $\|c_{i\ell}\| < c_{\max}$ is given in Assumption 1, we get $\sum_{(i,j)\in E(t)} h_{\rm ij}\|x_{\rm ij}\|^2 < N_a c_{\max}\varepsilon\sqrt{\varepsilon} \leq N c_{\max}\varepsilon\sqrt{\varepsilon}$, where we use the fact that $b_i = 0$, for $i \in \mathcal{N}_p$ and $\mathcal{N}_a \leq N$. Thus, $\forall\,(i,j) \in E(t)$, it holds that $h_{\rm ij}\|x_{\rm ij}\|^2 < N c_{\max}\varepsilon\sqrt{\varepsilon} \triangleq \varsigma$. It can be verified that $h_{\rm ij}\|x_{\rm ij}\|^2$ is monotonically increasing as a function of $\|x_{\rm ij}\|$. This implies that $\forall\,(i,j) \in E(t)$, $\|x_{\rm ij}\|^2 \leq r^2\varsigma$ or, equivalently, $\|x_{\rm ij}\|^2 \leq \varepsilon\sqrt{\varepsilon}\xi$, where $\xi \triangleq r^2 N c_{\max}$. Thus, $\|x_{\rm ij}\|$ can be made arbitrarily small by reducing ε. (II) Moreover, let ε_0 satisfy the condition

$$(N-1)\sqrt{\varepsilon_0\sqrt{\varepsilon_0}\xi} < r - \delta. \tag{14}$$

If $\varepsilon < \varepsilon_0$, then for *any* pair $(p,q) \in \mathcal{N}\times\mathcal{N}$, $\|x_{pq}\|$ satisfies $\|x_{pq}\| = |x_p - x_1 + x_1 - x_2 + \cdots - x_q| \leq (N-1)\sqrt{\varepsilon\sqrt{\varepsilon}\xi} < r - \delta$, as there exists a path in $G(t)$ of maximal length N from any node $p \in \mathcal{N}$ to another node q as $G(t)$ remains connected for $t > T_s$ by Theorem 1; and $\|x_{\rm ij}\| \leq \varepsilon\sqrt{\varepsilon}\xi$ from above, $\forall\,(i,j) \in E(t)$. By Definition 2, this implies $(p,q) \in E(t)$. Thus, $G(t)$ is a complete graph when $x(t) \in \mathcal{C}$. □

Namely, at the critical points, the graph $G(t)$ is complete, and thus remains invariant afterwards. Firstly, we define the following set for each active agent $i \in \mathcal{N}_a$:

$$\mathcal{S}_i \triangleq \left\{\mathbf{x} \in \mathbb{R}^{2N} \mid \|\mathbf{x} - \mathbf{1}_N \otimes c_{\rm ig}\| \leq r_S(\varepsilon)\right\} \tag{15}$$

where $r_S(\varepsilon) \triangleq \sqrt{3N\varepsilon} + \sqrt{(N-1)\varepsilon\sqrt{\varepsilon}\xi}$. Loosely speaking, $\mathcal{S}_i$ represents the neighbourhood around the goal region center of an active agent $i \in \mathcal{N}_a$. Furthermore, let $\mathcal{S} \triangleq \cup_{i\in\mathcal{N}_a}\mathcal{S}_i$ and $\mathcal{S}^{\neg} \triangleq \mathbb{R}^{2N} \setminus \mathcal{S}$. In the following, we analyze the properties of the critical points of $V(t)$ within the regions $\mathcal{S}$ and $\mathcal{S}^{\neg}$. More specifically: by Lemma 3 there are no local minima but saddle points within $\mathcal{S}^{\neg}$; by Lemma 4 these saddle points are non-degenerate; by Lemmas 5–7 all critical points within $\mathcal{S}$ are local minima. To explore these properties, we compute the second partial derivatives of $V(t)$ with respect to x_i:

$$\frac{\partial^2 V}{\partial x_i \partial x_i} = b_i d_i \otimes \mathbf{I}_2 + b_i d_i' p_i \cdot p_i^T + \sum_{j\in\mathcal{N}_i(t)} \left(h_{\rm ij} \otimes \mathbf{I}_2 + h_{\rm ij}' x_{\rm ij} \cdot x_{\rm ij}^T\right) \tag{16}$$

$$\frac{\partial^2 V}{\partial x_i \partial x_j} = -h_{\rm ij} \otimes \mathbf{I}_2 - h_{\rm ij}' x_{\rm ij} \cdot x_{\rm ij}^T, \qquad \forall\, j \neq i \tag{17}$$

where

$$d_i' = \frac{-4\varepsilon^3}{(\|p_i\|^2+\varepsilon)^3} + \frac{-\varepsilon^2}{(\|p_i\|^2+\varepsilon)^2}, \text{ and } h_{\rm ij}' = \frac{4r^2}{(r^2 - \|x_{\rm ij}\|^2)^3}.$$

Lemma 3: There are no local minima of V within $\mathcal{S}^{\neg}$.

Proof: We prove this by showing that if a critical point $\mathbf{x}_c \in \mathcal{S}^{\neg}$ there always exists a direction $\mathbf{z} \in \mathbb{R}^{2N}$ at $\mathbf{x}_c$ such that the quadratic form $\mathbf{z}^T\nabla^2 V\mathbf{z}$ is negative semi-definite. Given a critical point $\mathbf{x}_c \in \mathcal{C}$ and $\mathbf{x}_c \in \mathcal{S}^{\neg}$, then, by definition, $\|\mathbf{x} - \mathbf{1}_N \otimes c_{i\ell}\| > r_S(\varepsilon), \forall\, i \in \mathcal{N}_a$. Besides, for any $i \in \mathcal{N}_a$, we can bound $\|\mathbf{x} - \mathbf{1}_N \otimes c_{i\ell}\|$ as follows:

$$\|\mathbf{x} - \mathbf{1}_N \otimes c_{i\ell}\| = \|\mathbf{x} - \mathbf{1}_N \otimes x_i + \mathbf{1}_N \otimes (x_i - c_{\rm ig})\|$$

$$\leq \sqrt{\sum_{j\in\mathcal{N}} \|x_{\rm ij}\|^2} + \sqrt{N}\|p_i\| \leq \sqrt{(N-1)\varepsilon\sqrt{\varepsilon}\xi} + \sqrt{N}\|p_i\|$$

where we use the fact that $\|x_{\rm ij}\|^2 \leq \varepsilon\sqrt{\varepsilon}\xi$ at $\mathbf{x}_c, \forall\,(i,j) \in E(t)$ by Lemma 2. By comparing it with $r_S(\varepsilon)$ which is the lower bound, we get $\|p_i\| \geq \sqrt{3\varepsilon}, \forall\, i \in \mathcal{N}_a$. Choose $\mathbf{z} \triangleq \mathbf{1}_N \otimes z$, where $z \in \mathbb{R}^2$ and $\|z\| \triangleq 1$. Then, $\mathbf{z}^T\nabla^2 V\mathbf{z}$ is evaluated by using (16) and (17): $\mathbf{z}^T\nabla^2 V\mathbf{z} = \sum_{i\in\mathcal{N}} b_i d_i z^T z + b_i d_i' z^T p_i p_i^T z \triangleq z^T M z$, where $M \triangleq \sum_{i\in\mathcal{N}_a}(d_i \otimes \mathbf{I}_2 + d_i' p_i p_i^T)$ is a 2×2 Hermitian matrix, of which the trace is

$$\mathbf{trace}(M) = \sum_{i\in\mathcal{N}_a} 2d_i + d_i'\|p_i\|^2 = \varepsilon^3 \sum_{i\in\mathcal{N}_a} \frac{3\varepsilon - \|p_i\|^2}{(\|p_i\|^2+\varepsilon)^3} < 0$$

as we have shown that $\|p_i\| \geq \sqrt{3\varepsilon}$ above, $\forall\, i \in \mathcal{N}_a$ if $\mathbf{x}_c \in \mathcal{S}^{\neg}$. On the other hand, denote by $p_i = [p_{i,x}, p_{i,y}]$ the coordinates of p_i. The determinant of M is given by

$$\mathbf{det}(M) = -\left(\sum_{i\in\mathcal{N}_a} d_i' p_{i,x} p_{i,y}\right)^2 + \left(\sum_{i\in\mathcal{N}_a} d_i + d_i' p_{i,x}^2\right)\left(\sum_{i\in\mathcal{N}_a} d_i + d_i' p_{i,y}^2\right)$$

$$\geq \frac{1}{2}\sum_{i,j\in\mathcal{N}_a}\left[\left(d_i + d_i'\|p_i\|^2\right)\left(d_j + d_j'\|p_j\|^2\right)\right] > 0$$

since $d_i'\|p_i\|^2 < -d_i$ for $\|p_i\| > \sqrt{3\varepsilon}$, $\forall\, i \in \mathcal{N}_a$; and $(p_{i,x}p_{i,y} - p_{j,x}p_{j,y})^2 \leq \|p_i\|^2\|p_j\|^2$ by Cauchy-Schwarz inequality [13]. Denote by λ_1 and λ_2 the eigenvalues of M, where $\lambda_1, \lambda_2 \in \mathbb{R}$ as M is Hermitian. Since $\mathbf{trace}(M) < 0$ and $\mathbf{det}(M) > 0$, then M is negative definite and both eigenvalues are negative [13], i.e., $\lambda_1, \lambda_2 < 0$. Thus for any vector $z \in \mathbb{R}^2$, $z^T M z < 0$. In other words, for any vector $\mathbf{z} = \mathbf{1}_N \otimes z$ where $z \in \mathbb{R}^2$, $\mathbf{z}^T\nabla^2 V\mathbf{z} < 0$. To conclude, for any critical point $\mathbf{x}_c \in \mathcal{C}$, if $\mathbf{x}_c \in \mathcal{S}^{\neg}$ then $\mathbf{x}_c$ is not a local minimum. □

Lemma 4: There exists $\varepsilon_1 > 0$ such that if $\varepsilon < \varepsilon_1$, all critical points of V in $\mathcal{S}^{\neg}$ are non-degenerate saddle points.

Proof: To show that V is Morse, we use Lemma 3.8 from [27], which states that the non-singularity of a linear operator follows from the fact that its associated quadratic form is sign definite on complementary subspaces.

Let $\mathcal{Q} = \{v \in \mathbb{R}^{2N} | v = \mathbf{1}_N \otimes z, z \in \mathbb{R}^2\}$. In Lemma 3, we have shown that for any vector $v \in \mathcal{Q}$, $v^T\nabla^2 V v < 0$. Let $\mathcal{P} = \{v \in \mathbb{R}^{2N} | v = \mathbf{e}_N \otimes z, \mathbf{e}_N \perp \mathbf{1}_N, \mathbf{e}_N \in \mathbb{R}^N, z \in \mathbb{R}^2\}$. Firstly, it can be easily verified that $\mathcal{P}$ is the orthogonal complement of $\mathcal{Q}$. In the following, we show that $\nabla^2 V$ is positive definite in $\mathcal{P}$. Let $\mathbf{z} \in \mathcal{P}$, i.e., $\mathbf{z} \triangleq \mathbf{e}_N \otimes z \triangleq [z_1^T, z_2^T, \ldots, z_n^T]^T$, where $z \in \mathbb{R}^2$,

$\mathbf{e}_N \in \mathbb{R}^N$, $\mathbf{e}_N^T \perp \mathbf{1}_N$, $z_i \in \mathbb{R}^2, \forall\, i \in \mathcal{N}$. The quadratic form $\mathbf{z}^T \nabla^2 V \mathbf{z}$ at $\mathbf{x}_c$ can be computed explicitly using (16) and (17):

$$\begin{aligned}\mathbf{z}^T \nabla^2 V \mathbf{z} = &\sum_{i\in\mathcal{N}_a} \left(d_i\|z_i\|^2 + d_i' \left|p_i^T z_i\right|^2\right) \\ &+ \sum_{(i,j)\in E(t)} \Big(h_{\text{ij}}\|z_i - z_j\|^2 \\ &\qquad + 2h_{\text{ij}}' \left|(x_i - x_j)^T (z_i - z_j)\right|^2\Big) \\ \geq &\sum_{i\in\mathcal{N}_a} \left(d_i\|z_i\|^2 + d_i' \left|p_i^T z_i\right|^2\right) \\ &+ \sum_{(i,j)\in E(t)} h_{\text{ij}}\|z_i - z_j\|^2 \\ \geq &\sum_{i\in\mathcal{N}_a} \left(d_i + d_i'\|p_i\|^2\right)\|z_i\|^2 + \mathbf{z}^T(\mathbf{H}\otimes\mathbf{I}_2)\mathbf{z}\end{aligned}$$

where we use the fact that $h_{\text{ij}}' > 0$, $d_i' < 0$ and $|p_i^T z_i| \leq \|p_i\|\,\|z_i\|$. It can be verified that $d_i + d_i'\|p_i\|^2 > -0.1\varepsilon$ for $\|p_i\| \geq \sqrt{3\varepsilon}$, $\forall\, i \in \mathcal{N}_a$. Moreover, the second term can be lower-bounded by $\mathbf{z}^T(\mathbf{H}\otimes\mathbf{I}_2)\mathbf{z} = (\mathbf{e}_N \otimes z)^T \cdot (\mathbf{H}\otimes\mathbf{I}_2) \cdot (\mathbf{e}_N \otimes z) = (\mathbf{e}_N^T \cdot \mathbf{H} \cdot \mathbf{e}_N)\,\|z\|^2 \geq \lambda_2(\mathbf{H})\|\mathbf{z}\|^2$, where we apply the Courant-Fischer Theorem [13]: $\min_{\mathbf{e}_N \perp \mathbf{1}_N}\{\mathbf{e}_N^T \cdot \mathbf{H} \cdot \mathbf{e}_N\} = \lambda_2(\mathbf{H})\|\mathbf{e}_N\|^2 > 0$, since $\mathbf{H}$ is the Laplacian matrix defined in (13), which is positive semidefinite with $\lambda_1(\mathbf{H}) = 0$, of which the corresponding eigenvector is $\mathbf{1}_N$; and the second smallest eigenvalue $\lambda_2(\mathbf{H}) > 0$. In addition, since $h_{\text{ij}} > 1/r^2$ and $G(t)$ is a complete graph at $\mathbf{x}_c$ by Lemma 2, it holds that $\lambda_2(\mathbf{H}) > N/r^2$ by [8]. This implies that $\mathbf{z}^T\nabla^2 V\mathbf{z} \geq \sum_{i\in\mathcal{N}_a}((N/r^2) + d_i + d_i'\|p_i\|^2)\|z_i\|^2 \geq \sum_{i\in\mathcal{N}_a}((N/r^2) - 0.1\varepsilon)\|z_i\|^2$. Thus, if $\varepsilon < N/(0.1r^2)$, it holds that the quadratic form $\mathbf{z}^T\nabla^2 V\mathbf{z} > 0, \forall\, \mathbf{z} = \mathbf{e}_N \otimes z$ where $\mathbf{e}_N \perp \mathbf{1}_N$, $z \in \mathbb{R}^2$. To conclude, $\nabla^2 V|_{\mathcal{Q}}$ is negative definite by Lemma 3 and $\nabla^2 V|_{\mathcal{P}}$ is positive definite from the analysis above, given that ε satisfies the conditions below:

$$\varepsilon < \min\left\{\varepsilon_0, \frac{N}{0.1r^2}\right\} \triangleq \varepsilon_1. \tag{18}$$

By Lemma 3.8 from [27], we can conclude that $\nabla^2 V$ is nonsingular at the saddle points $\mathbf{x}_c \in \mathcal{S}^\neg$. Thus, all critical points within $\mathcal{S}^\neg$ are non-degenerate saddle points if $\varepsilon < \varepsilon_1$. □

Now, we focus on showing that all critical points within $\mathcal{S}$ are local minima. First, we need the following two lemmas stating that when the system is at a critical point belonging to $\mathcal{S}_i$ of any active agent $i \in \mathcal{N}_a$, then all the other agents are within this goal region π_{ig} and away from their own goal region center by at least distance $r_{\min}$.

Lemma 5: There exists $\varepsilon_2 > 0$ that if $\varepsilon < \varepsilon_2$, the following statements hold: (I) $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset, \forall\, i \neq j$ and $i, j \in \mathcal{N}_a$; (II) If $\mathbf{x}_c \in \mathcal{S}_i$ for any $i \in \mathcal{N}_a$, then $x_j \in \pi_{\text{ig}}, \forall\, j \in \mathcal{N}$ and $\|x_j - c_{j\text{g}}\| > r_{\min}$, $j \neq i, \forall\, j \in \mathcal{N}_a$.

Proof: Let ε_2 be given as the solution of

$$r_S(\varepsilon_2) \triangleq \sqrt{3N\varepsilon_2} + \sqrt{(N-1)\varepsilon_2\sqrt{\varepsilon_2}\xi} \triangleq r_{\min} \tag{19}$$

where $r_{\min}$ is given in Assumption 1. Note that ε_2 is unique as the left-hand side is a function of ε_2 that monotonically increases and has the range $[0, \infty)$. Assume that $\mathbf{x}_c \in \mathcal{S}_{i^\star}$ for some $i^\star \in \mathcal{N}_a$, i.e., $\|\mathbf{x}_c - \mathbf{1}_N \otimes c_{i^\star\text{g}}\| \leq r_S(\varepsilon_2)$. Then, $\forall\, j \neq i^\star$, $j \in \mathcal{N}_a$, it holds that (I) $\|\mathbf{x}_c - \mathbf{1}_N \otimes c_{j\text{g}}\| = \|\mathbf{x}_c - \mathbf{1}_N \otimes c_{i\text{g}} + \mathbf{1}_N \otimes c_{i\text{g}} - \mathbf{1}_N \otimes c_{j\text{g}}\| \geq \sqrt{N}\|c_{i\text{g}} - c_{j\text{g}}\| - \|\mathbf{x}_c - \mathbf{1}_N \otimes c_{i\text{g}}\| \geq 2\sqrt{N}r_{\min} - r_S(\varepsilon)$, due to that $\|c_{\text{ig}} - c_{j\text{g}}\| > 2r_{\min}$ by Assumption 1. Since $\varepsilon < \varepsilon_2$, then $r_S(\varepsilon) < r_S(\varepsilon_2) = r_{\min}$. Thus, $\|\mathbf{x}_c - \mathbf{1}_N \otimes c_{j\text{g}}\| > 2\sqrt{N}r_{\min} - r_{\min} > r_{\min} = r_S(\varepsilon_2)$, implying that $\mathbf{x}_c \notin \mathcal{S}_j$. (II) $\|x_j - c_{i^\star\text{g}}\| < \|\mathbf{x}_c - \mathbf{1}_N \otimes c_{i^\star\text{g}}\| < r_{\min} < r_{i^\star\text{g}}$, meaning that $x_j \in \pi_{i^\star\text{g}}, \forall\, j \in \mathcal{N}$. Thus, for each active agent $j \in \mathcal{N}_a$, it holds $\|x_j - c_{j\text{g}}\| = \|x_j - c_{i^\star\text{g}} + c_{i^\star\text{g}} - c_{j\text{g}}\| \geq \|c_{i^\star\text{g}} - c_{j\text{g}}\| - \|x_j - c_{i^\star\text{g}}\| \geq 2r_{\min} - r_{\min} > r_{\min}$. □

Lemma 6: There exists $\varepsilon_6 > 0$ such that if $\varepsilon < \varepsilon_6$, then for any critical point $\mathbf{x}_c \in \mathcal{S}_i$, $i \in \mathcal{N}_a$, then it holds that $\|p_i\| < \sqrt{0.4\varepsilon}$.

Proof: Without loss of generality, let $\mathbf{x}_c \in \mathcal{S}_{i^\star}$, where $i^\star \in \mathcal{N}_a$. By summing (12) for all $i \in \mathcal{N}$, we get $d_{i^\star}p_{i^\star} = -\sum_{j\neq i^\star, j\in\mathcal{N}_a} d_j p_j$. Consider the scalar function $f(\|p_j\|) = d_j(\|p_j\|)\|p_j\|$ for $\|p_j\| \geq 0$. It is monotonically increasing for $\|p_j\| \in [0, 3.2\sqrt{\varepsilon})$ and decreasing for $\|p_j\| \in [3.2\sqrt{\varepsilon}, \infty)$. If $\mathbf{x}_c \in \mathcal{S}_{i^\star}$ for $i^\star \in \mathcal{N}_a$, then $\|\mathbf{x}_c - \mathbf{1}_N \otimes c_{i^\star\text{g}}\| \leq r_S(\varepsilon_2)$. Moreover, $\|\mathbf{x} - \mathbf{1}_N \otimes c_{i^\star\ell}\| \geq \|\mathbf{1}_N \otimes x_{i^\star} - \mathbf{1}_N \otimes c_{i^\star\text{g}}\| - \|\mathbf{x} - \mathbf{1}_N \otimes x_{i^\star}\| \geq \sqrt{N}\|p_{i^\star}\| - \sqrt{(N-1)\varepsilon\sqrt{\varepsilon}\xi}$. This implies $\|p_{i^\star}\| \leq \sqrt{3\varepsilon} + 2\sqrt{\varepsilon\sqrt{\varepsilon}\xi}$. Moreover by Lemma 5, $\|p_j\| > r_{\min}, \forall\, j \neq i^\star$, $j \in \mathcal{N}_a$. Thus, if $r_{\min} > 3.2\sqrt{\varepsilon}$, namely, $\varepsilon < 0.07r_{\min}^2 \triangleq \varepsilon_3$, it holds that $d_j\|p_j\| < 0.5\varepsilon^2/r_{\min}, \forall\, j \neq i^\star$, $j \in \mathcal{N}_a$. Thus, $d_{i^\star}\|p_{i^\star}\| < 0.5(N_a - 1)\varepsilon^2/r_{\min}$. If the following two conditions hold: (i) $\sqrt{3\varepsilon} + 2\sqrt{\varepsilon\sqrt{\varepsilon}\xi} < 3.2\sqrt{\varepsilon}$; (ii) $0.5(N_a - 1)\varepsilon^2/r_{\min} < d_j(\sqrt{0.4\varepsilon})\sqrt{0.4\varepsilon}$, then $\|p_{i^\star}\| < \sqrt{0.4\varepsilon}$ since it is shown earlier that function $d_j(\|p_j\|)\|p_j\|$ is monotonically increasing for $\|p_j\| \in [0, 3.2\sqrt{\varepsilon})$. Condition (i) above implies that $\varepsilon < 4.1/\xi^2 \triangleq \varepsilon_4$ and condition (ii) holds for any $N_a \leq N$ if $\varepsilon < 0.8r_{\min}^2/(N-1)^2 \triangleq \varepsilon_5$. To conclude, if $\varepsilon < \varepsilon_6$, where

$$\varepsilon_6 \triangleq \min\{\varepsilon_3, \varepsilon_4, \varepsilon_5\} \tag{20}$$

then $\mathbf{x}_c \in \mathcal{S}_{i^\star}$ implies $\|p_{i^\star}\| < \sqrt{0.4\varepsilon}$. □

With the above two lemmas, we can now show that all critical points of $V(t)$ within $\mathcal{S}$ are local minima.

Lemma 7: There exists $\varepsilon_{\min} > 0$ such that if $\varepsilon < \varepsilon_{\min}$, all critical points of V within $\mathcal{S}$ are local minima.

Proof: A critical point $\mathbf{x}_c \in \mathcal{S}$ can only belong to one set $\mathcal{S}_i$ of an active agent $i \in \mathcal{N}_a$ by Lemma 5. Let $\mathbf{x}_c \in \mathcal{S}_{i^\star}$, where $i^\star \in \mathcal{N}_a$. Let $\mathbf{z} \in \mathbb{R}^{2N}$ and $\|\mathbf{z}\| = 1$. Set $\mathbf{z} = [z_1^T, z_2^T, \ldots, z_n^T]^T$, where $z_i \in \mathbb{R}^2, \forall\, i \in \mathcal{N}$. Then, $\mathbf{z}^T\nabla^2 V\mathbf{z}$ at $\mathbf{x}_c$ is computed as:

$$\begin{aligned}\mathbf{z}^T\nabla^2 V\mathbf{z} = &\sum_{i\in\mathcal{N}_a}\left(d_i\|z_i\|^2 + d_i'\left|p_i^T z_i\right|^2\right) \\ &+ \sum_{(i,j)\in E(t)}\left(h_{\text{ij}}\|z_{\text{ij}}\|^2 + 2h_{\text{ij}}'\left|x_{\text{ij}}^T z_{\text{ij}}\right|^2\right).\end{aligned} \tag{21}$$

where $z_{\text{ij}} \triangleq z_i - z_j$. Since $|p_i^T z_i| \leq \|p_i\|\|z_i\|$, $d_i > 0$ and $d_i' < 0$, it holds that $d_i\|z_i\|^2 + d_i'|p_i^T z_i|^2 \geq (d_i + d_i'\|p_i\|^2)\,\|z_i\|^2, \forall\, i \in \mathcal{N}_a$. It holds that for $j \neq i^\star$ and $\forall\, j \in \mathcal{N}_a$, $d_j + d_j'\|p_j\|^2 > \varepsilon^2\hat{g}$, where $\hat{g} \triangleq -2/r_{\min}^2$, since $\|p_j\| > r_{\min}$ by Lemma 5; and $d_{i^\star} + d_{i^\star}'\|p_{i^\star}\|^2 > 0.08\varepsilon$ since $\|p_{i^\star}\| > \sqrt{0.4\varepsilon}$ by Lemma 6. Regarding the second term of (21), since Lemma 2 shows that $G(t)$ is a complete graph at $\mathbf{x}_c$ with $h_{\text{ij}} > 1/r^2$ and

$h'_{\rm ij} > 0$, we get $\sum_{(i,j)\in E}(h_{\rm ij}\|z_{\rm ij}\|^2 + 2h'_{\rm ij}|x_{\rm ij}^T z_{\rm ij}|^2) \geq \sum_{j\in\mathcal{N}} \|z_{i^\star j}\|^2/r^2$. Thus, (21) can be bounded by

$$\begin{aligned}\mathbf{z}^T\nabla^2 V\mathbf{z} &\geq \sum_{i\in\mathcal{N}_a}\left(d_i + d'_i\|p_i\|^2\right)\|z_i\|^2 + \sum_{j\in\mathcal{N}} h_{i^\star j}\|z_{i^\star j}\|^2\\ &\geq 0.08\varepsilon\|z_{i^\star}\|^2 - \varepsilon^2\sum_{j\neq i^\star, j\in\mathcal{N}_a}|\hat{g}|\|z_j\|^2\\ &\quad + \frac{1}{r^2}\sum_{j\in\mathcal{N}}\|z_{i^\star j}\|^2\\ &\geq \sum_{j\in\mathcal{N}_a}\left(\frac{1}{r^2} + \frac{0.08\varepsilon}{N}\right)\|z_{i^\star}\|^2\\ &\quad + \left(\frac{1}{r^2} - \varepsilon^2|\hat{g}|\right)\|z_j\|^2 - \frac{2}{r^2}z_{i^\star}^T z_j\end{aligned}$$

as $1 \leq N_a \leq N$. If the following condition holds: $((1/r^2) + (0.08\varepsilon/N))((1/r^2) - \varepsilon^2|\hat{g}|) > (1/r^2)^2$, it implies $\mathbf{z}^T\nabla^2 V\mathbf{z} > (|z_{i^\star}^T z_j| - z_{i^\star}^T z_j)/r^2 \geq 0, \forall\, \mathbf{z}\in\mathbb{R}^{2N}$, i.e., $\nabla^2 V$ is positive definite at $\mathbf{x}_c \in \mathcal{S}$. The above condition is equivalent to $\varepsilon^2 + (N/0.08r^2)\varepsilon - (1/r^2|\hat{g}|) < 0$. Since $\varepsilon > 0$, this implies that

$$0 < \varepsilon < \frac{\sqrt{\left(\frac{N}{0.08r^2}\right)^2 + \frac{4}{r^2|\hat{g}|}} - \frac{N}{0.08r^2}}{2} \triangleq \varepsilon_7. \qquad (22)$$

To conclude, if

$$\varepsilon < \min\{\varepsilon_1, \varepsilon_2, \varepsilon_6, \varepsilon_7\} \triangleq \varepsilon_{\min} \qquad (23)$$

where ε_1, ε_2, ε_6 and ε_7 are positive and defined in (18)–(20) and (22), then all local minima within $\mathcal{S}$ are stable. □

By summarizing Lemmas 3–7, we can derive the following convergence result for the controlled closed-loop system:

Theorem 8: Assume that $G(T_s)$ is connected and $\varepsilon < \varepsilon_{\min}$ by (23). Then, starting from anywhere in the workspace except a set of measure zero, there exists a finite time $T_f \in [T_s, \infty)$ and one agent $i^\star \in \mathcal{N}_a$, such that $x_j(T_f) \in \pi_{i^\star \rm g}, \forall\, j \in \mathcal{N}$, while at the same time $\|x_i(t) - x_j(t)\| < r, \forall\,(i,j)\in E(T_s)$ and $\forall\, t \in [T_s, T_f]$.

Proof: Firstly, the second part follows from Theorem 1, which guarantees that all edges within $E(T_s)$ will be reserved for all $t > T_s$. Secondly, we have shown that $V(t)$ by (7) is lower-bounded and non-increasing after $G(t)$ becomes complete by Lemma 2. By LaSalle's invariance principle [16], we only need to find out the largest invariant set within $\dot{V}(t) = 0$. Lemmas 3 to 7 ensure that $V(t)$ has only local minima inside $\mathcal{S}$ and saddle points outside $\mathcal{S}$. These saddle points have attractors of measure zero by Lemma 4. Thus, starting from anywhere in the workspace except a set of measure zero, the system converges to the set of local minima. Part (I) of Lemma 5 shows that a local minimum can not belong to two different $\mathcal{S}_i$ simultaneously. Thus, the system converges to the set of local minima within $\mathcal{S}_{i^\star}$ for one active agent $i^\star \in \mathcal{N}_a$. By part (II) of Lemma 5, all agents would be inside $\pi_{i^\star \rm g}$ at a critical point within $\mathcal{S}_{i^\star}$, i.e., $x_j \in \pi_{i^\star \rm g}, \forall\, j\in\mathcal{N}$. Consequently, by continuity, there exists a finite time $T_f < \infty$ that $x_j(T_f) \in \pi_{i^\star \rm g}, \forall\, j \in \mathcal{N}$, for one active agent $i^\star \in \mathcal{N}_a$. □

Remark 1: Note that Theorem 8 above holds for any number of active agents that $1 \leq N_a \leq N$. Namely, independent of the number of active agents, one of the active agents will reach its goal region first within finite time, while the whole team fulfills the relative-distance constraints at all time.

C. Hybrid Control Structure

In this part, we propose *three* different switching protocols for each agent to decide on its own activity or passivity under there different cases, such that all agents can fulfill their local tasks and at the same time satisfy the relative-distance constraints. Through these protocols, we can integrate the discrete plan execution from Section IV-A and the continuous control laws from Section IV-B into a hybrid control scheme, which monitors the plan execution and motion control online and in real-time. This hybrid scheme is fully decentralized and only relies on local relative-state measurements.

1) Switching Protocol for sc-LTL: Let us first focus on a case where each local task φ_i, $i\in\mathcal{N}$ is an sc-LTL formula. As introduced in Section IV-A, the discrete plan τ_i for agent i is a finite satisfying prefix of progressive goal regions and the set of services to provide at each region: $\tau_{i,\rm pre} = (\pi_{i1}, w_{i1})\cdots(\pi_{ik_i}, w_{ik_i})$, where $\pi_{i1}, \pi_{i2}, \ldots, \pi_{ik_i} \in \Pi_i$ and $w_{i1}, w_{i2}, \ldots, w_{ik_i} \in 2^{\Sigma_i}$. We propose the following *activity switching protocol* for each agent $i\in\mathcal{N}$, (referred by $\mathbf{P_{sc}}$):

(1) At time $t = 0$, agent i sets $\varkappa_i := 1$ and itself as active and sets $\pi_{\rm ig} := \pi_{i\varkappa_i}$, namely, the first goal region by τ_i. The *active* controller (5) is applied to agent i, where the progressive goal region is $\pi_{\rm ig}$, i.e., $c_{\rm ig} = c_{i\ell_1}$.
(2) Whenever agent i reaches its current progressive goal region $\pi_{\rm ig} = \pi_{i\varkappa_i}$ and $\varkappa_i < k_i$, it provides the prescribed set of services $w_{i\varkappa_i}$ by τ_i and it sets $\varkappa_i := \varkappa_i + 1$ and $\pi_{\rm ig} := \pi_{i\varkappa_i}$. Then, the controller (5) for agent i is updated accordingly by setting $c_{\rm ig} = c_{i\ell_{k+1}}$.
(3) Whenever agent i reaches its last progressive goal region $\pi_{\rm ig} = \pi_{ik_i}$, it provides the set of services w_{ik_i} by which it finishes the execution of its finite discrete plan τ_i. Afterward, it remains *passive* by controller (6).

Theorem 9: By the protocol $\mathbf{P_{sc}}$ above, it is guaranteed that $\forall\, i\in\mathcal{N}$, φ_i is satisfied by $\mathbf{x}_i(T)$, and $\|x_i(t) - x_j(t)\| < r$, $\forall\,(i,j)\in E_0(0)$ and $\forall\, t\geq 0$, where $T = \infty$.

Proof: At $t = 0$, all agents are active and following the controller (5). By Theorem 8, all agents converge to one agent's goal region at a finite time $t_1 > 0$. Denote by $i\in\mathcal{N}$ this agent. Then, either by step (II) of the protocol agent i updates its active control law by setting $\pi_{\rm ig} = \pi_{i2}$, or by step (III) agent i has completed its plan $\tau_{i,\rm pre}$ and becomes passive. Since all agents' plans are finite and Theorem 8 holds for any number of active agents, we obtain that there exists a finite time instant T_{f_j}, such that one of the agents $j\in\mathcal{N}_a$ finishes executing its plan $\tau_{j,\rm pre}$, i.e., such that φ_j becomes satisfied. Then, by step (III), this agent is passive by controller (6) for all times $t \in [T_{f_j}, \infty)$. Inductively, we conclude that there exists a time instant T_f, by which all agents complete their plans. All agents are passive for all $t \in (T_f, \infty)$ and by controller (6) they all converge to one point. The second part of the theorem follows directly from Theorem 8. □

2) Switching Protocol for General LTL: As introduced in Section IV-A, if the task specification φ_i is given as a general LTL formula, then the plan τ_i is given by an infinite sequence in the prefix-suffix form: $\tau_i = \tau_{i,\text{pre}}(\tau_{i,\text{suf}})^\omega = (\pi_{i1}, w_{i1})(\pi_{i2}, w_{i2})\ldots$, where $\tau_{i,\text{pre}} = (\pi_i, w_{i1})\ldots(\pi_{ik_i}, w_{ik_i})$, for $k_i > 0$ and $\tau_{i,\text{suf}} = (\pi_{ik_i+1}, w_{ik_i+1})\ldots(\pi_{iK_i}, w_{iK_i})$, where $\pi_{i1}, \pi_{i2}, \ldots, \pi_{iK_i} \in \Pi_i$ is the sequence of goal regions and $w_{i1}, w_{i2}, \ldots, w_{iK_i} \in 2^{\Sigma_i}$ is the associated sequence of services.

The main challenge in this case is to ensure that each agent executes its plan suffix infinitely often. The activity switching protocol $\mathbf{P_{sc}}$ from Section IV-C1 could not be applied here since all agents should remain active at all time due to the infinite discrete plan. Besides, it is possible that the team may repetitively converge to π_{ig} for one agent $i \in \mathcal{N}$ while never visiting the other agents' progressive goal regions. Hence, we aim here to design a "fair" protocol that enforces a progressive satisfaction towards each agent's local task.

Reaching-Event Detector: Agent $i \in \mathcal{N}$ can detect when it reaches its own progressive goal region π_{ig} by checking if $x_i(t) \in \pi_{\text{ig}}$. However, it is also essential that it can detect when another agent $j \in \mathcal{N}$ reaches $\pi_{j\text{g}}$. Since the connectivity graph is complete, it is sufficient for agent i to detect when a neighboring agent $j \in \mathcal{N}_i(t)$ reaches $\pi_{j\text{g}}$. Given that the agents satisfy the dynamics by (1) and that each agent $i \in \mathcal{N}$ can measure $x_i(t) - x_j(t), \forall\, j \in \mathcal{N}_i(t)$ in real time, we assume that the agent i can measure or estimate [6] $u_j(t), \forall\, j \in \mathcal{N}_i(t)$. Let $\Omega_i(j,t) \in \mathbb{B}$ be a Boolean variable indicating that agent i detects its neighboring agent $j \in \mathcal{N}_i(t)$ reaching the goal region $\pi_{j\text{g}}$ at time $t > 0$. We propose a reaching-event detector below inspired by [25]. Simply speaking, the detector checks if within a short time period $[t - \Delta_t, t]$, there exists $j \in \mathcal{N}_i(t)$, such that $u_j(t)$ has changed from a relatively small value (below a given Δ_u) by a difference larger than certain Δ_d. If so, it indicates that the agent j has reached its progressive goal region $\pi_{j\text{g}}$. This design is motivated by the following facts: By Theorem 7, the system is at a local minimum whenever an active agent is in its progressive goal region. Thus, when the agent j reaches $\pi_{j\text{g}}$ at time t, all control inputs $u_i(t)$ are close to zero for all $i \in \mathcal{N}$ by (12). Afterward, our switching protocol guarantees that *only* agent j switches its control law either to (5) to navigate to the next goal region or to (6) to become passive. This change is lower-bounded by Δ_d derived using control law (5) and Lemmas 5 and 6 as $\Delta_d \triangleq |f(r_{\min}) - f(\sqrt{0.4\varepsilon})|$, where $f(\|p_j\|) = d_j(\|p_j\|)\|p_j\|$ is a scalar function and $d_j(\|p_j\|)$ is defined by (Section IV-B). In contrast, for the other agents $i \neq j$, $i \in \mathcal{N}$, the control input $u_i(t)$ remains unchanged and close to zero.

Activity Switching Protocol: Firstly, we define a *round* as the time period during which each agent has reached at least one of its goal regions according to their plans.

Definition 3: For all $m \geq 1$, the *m-th round* is defined as the time interval $[T_{\circlearrowleft_{m-1}}, T_{\circlearrowleft_m})$, where $T_{\circlearrowleft_0} = 0$, $T_{\circlearrowleft_{m-1}} < T_{\circlearrowleft_m}$ and for all $m \geq 1$, $T_{\circlearrowleft_m}$ is the smallest time satisfying the following conditions for all $i \in \mathcal{N}$: $\texttt{word}_i(T_{\circlearrowleft_m}) = w_{i1}w_{i2}\cdots w_{i\ell}$ for some $\ell \geq 1$ and $\texttt{word}_i(T_{\circlearrowleft_m}) \neq \texttt{word}_i(T_{\circlearrowleft_{m-1}})$.

This notion of a round is crucial to the protocol design below. We firstly introduce two local variables: $\chi_i \geq 0$ that indicates the starting time of the current round and $\Upsilon_i \in \mathbb{Z}^N$ a vector to record how many progressive goal regions each agent has reached within one round since χ_i. Then, the *activity switching protocol* (referred by $\mathbf{P_{ge}}$) is as follows:

(1) At time $t = 0$, $\Upsilon_i := \mathbf{0}_N$, $\chi_i := 0$, $\varkappa_i := 1$. The agent i is active and follows control law (5), where $\pi_{\text{ig}} := \pi_{i\varkappa_i}$.

(2) Whenever the agent i reaches its current progressive goal region $\pi_{\text{ig}} = \pi_{i\varkappa_i}$, it provides the prescribed set of services $w_{i\varkappa_i}$ by τ_i and updates the current progressive goal region accordingly: If $\varkappa_i < K_i$ then $\varkappa_i := \varkappa_i + 1$, and if $\varkappa_i = K_i$ then $\varkappa_i := k_i + 1$. Furthermore, $\pi_{\text{ig}} := \pi_{i\varkappa_i}$, and finally $\Upsilon_i[i] := \Upsilon_i[i] + 1$. Generally speaking, the agent i decides to stay active or to become passive based on the probability function:

$$\mathbf{Pr}(b_i = 1) = \begin{cases} f_{\text{prob}}(\cdot) & \text{if } f_{\text{cond}}(\cdot) = \texttt{True} \\ 0 & \text{otherwise} \end{cases}$$

where $f_{\text{prob}}(\cdot) \in [0,1]$ and $f_{\text{cond}}(\cdot) \in \{\texttt{True}, \texttt{False}\}$ are functions of time t and the local variables Υ_i and χ_i, subject to the following: given that the current round is the m-th one, there exists a time $T \in (T_{\circlearrowleft_{m-1}}, T_{\circlearrowleft_m})$, such that $f_{\text{cond}}(\cdot) = \texttt{False}$ for all $t \in [T, T_{\circlearrowleft_m})$. Whenever $b_i = 1$, the agent i keeps following the control law (5) with the updated π_{ig}. Otherwise, it becomes passive and the control law (6) is applied.

(3) Whenever agent i detects that $\Omega_i(j,t) = \texttt{True}$, for some $j \neq i \in \mathcal{N}$, it sets $\Upsilon_i[j] = \Upsilon_i[j] + 1$.

(4) Whenever $\Upsilon_i[j] \geq 1, \forall\, j \in \mathcal{N}$, i.e., all elements of Υ_i are positive, then agent i sets $\Upsilon_i := \mathbf{0}_N$, $\chi_i := t$ and follows the active control law (5) to its goal region π_{ig}.

A straightforward choice of the function $\mathbf{Pr}(\cdot)$ is $f_{\text{cond}} = \texttt{False}$, for all $t \geq 0$. Then, the agent i always becomes passive once it visits π_{ig} and it becomes active again after the current round is completed by step (IV). In this case, the number of active agents gradually decreases within each round. However, a different choice may allow trading the fairness of activity switching with the increased efficiency of plan executions. The switching to passive control mode may be temporarily postponed and, thus, the visits to progressive goal regions may become more frequent. Examples are given in Section V.

Lemma 10: The round $[T_{\circlearrowleft_{m-1}}, T_{\circlearrowleft_m})$ is finite, $\forall\, m \geq 1$.

Proof: Let $t = T_{\circlearrowleft_{m-1}} = 0$ and, thus, $\Upsilon_i[j] = 0$, for all $i, j \in \mathcal{N}$ by step (I). By Theorem 8, one of the agents reaches its progressive goal region in finite time at $t_1 \geq T_{\circlearrowleft_{j-1}}$. Since there are only finite number of agents and due to the required properties of f_{cond}, there exists a finite time $T_{f_j} \geq 0$, when either the step (IV) applies or when one of the agents $j \in \mathcal{N}_a$ necessarily becomes passive by the function $\mathbf{Pr}(\cdot)$ in step (II) and remains passive till the end of the round. In the former case, $T_{\circlearrowleft_m} = T_{f_j}$, i.e., we directly obtain that the first round is finite. In the latter case, the same argument can be applied to the $N-1$ active agents such that one of them will become passive in finite time. By repeating this process, we obtain that there exists a finite time instant T_f, such that step (IV) applies, i.e., such that $T_{\circlearrowleft_m} = T_f$. Again, we derive that the first round is finite. Inductively, let $m > 1$, $t = T_{\circlearrowleft_{m-1}}$, and $\Upsilon_i[j] = 0$, for all $i, j \in \mathcal{N}$ by step (IV). Using analogous arguments as above, we derive that the m_{th} round $[T_{\circlearrowleft_m}, T_{\circlearrowleft_{m+1}}]$ is finite. □

Theorem 11: By the protocol $\mathbf{P_{ge}}$ above, it is guaranteed that $\forall\, i \in \mathcal{N}$, φ_i is satisfied by $\mathbf{x}_i(T)$ and $\|x_i(t) - x_j(t)\| < r$, $\forall\,(i,j) \in E_0(0)$ and $\forall\, t > 0$, where $T = \infty$.

Proof: (Sketch) The satisfaction of φ_i follows directly from the correctness of each agent's discrete plan and the fact that each round is finite by Lemma 10. At last, the relative-distance constraints are always maintained as shown in Theorem 8. □

3) Switching Protocol for Mixed Task Specifications: As stated in Section III, the task specifications $\{\varphi_i, i \in \mathcal{N}\}$ can be of different types. Namely, some tasks are given as sc-LTL formulas (denoted by $\mathcal{N}_{\text{sc}} \subseteq \mathcal{N}$) and some are given as general LTL formulas (denoted by $\mathcal{N}_{\text{ge}} \subseteq \mathcal{N}$).

Firstly, we show that a new switching protocol is needed for this case, i.e., simply applying the protocol $\mathbf{P_{sc}}$ for agents in $\mathcal{N}_{\text{sc}}$ and the protocol $\mathbf{P_{ge}}$ for agents in $\mathcal{N}_{\text{ge}}$ is not a valid solution. The reason is that when one agent $j \in \mathcal{N}_{\text{sc}}$ has finished executing its plan, it would switch to being passive indefinitely by step (III) of $\mathbf{P_{sc}}$. After that, for all agents $i \in \mathcal{N}_{\text{ge}}$, one round may never finish in finite time since step (IV) of $\mathbf{P_{ge}}$ will not be reached as $\Upsilon_i[j] = 0$ holds always. To that end, we firstly propose another event detector as follows:

All-Passive Detector: Let $\Psi_i(t) \in \mathbb{B}$ be a Boolean variable which indicates that agent $i \in \mathcal{N}_{\text{ge}}$ detects that *all* of its neighboring agents are in the passive mode at time $t > 0$. As discussed earlier, when all agents within $\mathcal{N}$ are passive and following the controller (6), the closed-loop system dynamics can be described by $\dot{\mathbf{x}} = -(\mathbf{H} \otimes I_2)\mathbf{x}$, where the matrix $\mathbf{H}$ is defined in Section IV-B2. It has been shown that $\mathbf{H}$ is positive semidefinite with only one zero eigenvalue. As a result, all agents would asymptotically converge to one rendezvous point [26]. In other words, $x_{\text{ij}}(t) = x_i(t) - x_j(t) \to 0$ and $u_i(t) \to 0$ as $t \to +\infty$, $\forall\,(i,j) \in \mathcal{N} \times \mathcal{N}$. Thus, we propose that $\Psi_i(t')$ becomes *True* if agent i detects that $|u_j| < \Delta_c$ holds, $\forall\, j \in \mathcal{N}_i$ and $\forall\, t \in [t' - \Delta_p, t']$, where $\Delta_c > 0$ is the upper bound on the control input and $\Delta_p > 0$ is the monitoring period. Given the appropriately chosen Δ_c and Δ_p, $\Psi_i(t')$ becomes *True* only when all agents are passive at time $t = t'$. Without loss of generality, assume there is at least one agent being active in the team at time $t = t'$. Since $|u_i| < \Delta_c$ holds for all time $t \in [t' - \Delta_p, t']$, it means the system stays at the critical point of $V(t)$ associated with one of the active agents for at least the time interval $[t' - \Delta_p, t']$. By the analysis of $V(t)$ from Section IV-B, this violates the fact that all active agents should navigate to their individual goal regions by following (5). Thus, $\Psi_i(t)$ becomes *True* only when all agents are passive at time t.

Then, the *activity switching protocol* for this case, denoted by $\mathbf{P_{mx}}$, is designed as follows: for any agent $i \in \mathcal{N}_{\text{sc}}$, it simply follows the switching protocol $\mathbf{P_{sc}}$. Namely, it traverses the sequences of goal regions and provides the set of services there according to its finite plan $\tau_{i,\text{pre}}$. After it finishes the execution, it remains passive indefinitely. On the other hand, for any agent $i \in \mathcal{N}_{\text{ge}}$, we introduce a new variable $\mathcal{N}_{i,\text{sc}}(t) \subseteq \mathcal{N}_i(t)$ to save the set of agent i's neighbors belonging to $\mathcal{N}_{\text{sc}}$, which is initialized as empty and maintained locally by agent i. For any agent $i \in \mathcal{N}_{\text{ge}}$, the steps (I)–(III) of protocol $\mathbf{P_{ge}}$ remain the same, but (IV) should be modified as follows, and an additional step (V) needs to be added:

(IV) Whenever it holds that $\Upsilon_i[j] \geq 1, \forall\, j \in \mathcal{N}$ and $j \notin \mathcal{N}_{i,\text{sc}}$, then agent i sets $\Upsilon_i := \mathbf{0}_N$, $\chi_i := t$ and follows the active control law (5) to its goal region π_{ig}.

(V) Whenever agent i detects that $\Psi_i(t) = \texttt{True}$, then $\forall\, j \in \mathcal{N}$, if $\Upsilon_i[j] = 0$, add j to $\mathcal{N}_{i,\text{sc}}$.

Namely, by step (IV) above, each agent $i \in \mathcal{N}_{\text{ge}}$ would reset Υ_i to 0 and start a new round once every agent has made a progress in its plan execution, *except* those belonging to $\mathcal{N}_{i,\text{sc}}$.

Then, when $\Psi_i(t) = \texttt{True}$, it means that all agents are in the passive mode. If the neighbor $j \in \mathcal{N}_i$ also belongs to $\mathcal{N}_{\text{ge}}$, by step (II) of protocol $\mathbf{P_{ge}}$ in Section IV-C2 agent j must have reached its goal region at least once, i.e., $\Upsilon_i[j] \geq 1$. Thus if $\Upsilon_i[j] = 0$ for some neighbor $j \in \mathcal{N}_i$ when $\Psi_i(t) = \texttt{True}$, it implies $j \in \mathcal{N}_{\text{sc}}$ and moreover agent j has finished executing its finite plan according to step (II) of protocol $\mathbf{P_{sc}}$, and it remains passive afterwards with $\Upsilon_i[j]$ being constantly zero. Thus agent i adds j to $\mathcal{N}_{i,\text{sc}}$ by step (V) above.

Theorem 12: By the protocol $\mathbf{P_{mx}}$ above, it is guaranteed that $\forall\, i \in \mathcal{N}$, φ_i is satisfied by $\mathbf{x}_i(T)$ and $\|x_i(t) - x_j(t)\| < r$, $\forall\,(i,j) \in E_0(0)$ and $\forall\, t > 0$, where $T = \infty$.

Proof: Similar to Theorem 11, we only need to show that in this case one "round" is also finite. Note that now the definition of round differs slightly from Definition 3 as here it is only defined for all agents in $\mathcal{N}_{\text{ge}}$. Before any agent $j \in \mathcal{N}_{\text{sc}}$ finishes executing its plan, one round is clearly finite as it ends once eve ry agent has reached at least one of its progressive goal regions. Consider that one or more agents within $\mathcal{N}_{\text{sc}}$ (denoted by $\mathcal{N}_1 \subseteq \mathcal{N}_{\text{sc}}$) have finished executing their plans and become passive. All agents within $\mathcal{N}_{\text{ge}}$ will reach their goal regions at least once before they become passive in finite time as shown in Lemma 10, i.e., $\Upsilon_i[j] \geq 1, \forall\, j \in \mathcal{N}_{\text{ge}}$, while the agents in $\mathcal{N}_1$ remain passive since last round, i.e., $\Upsilon_i[j] = 0, \forall\, j \in \mathcal{N}_1$. According to step (V) above, each agent $i \in \mathcal{N}_{\text{ge}}$ would detect that all agents are passive and add all agents in $\mathcal{N}_1$ to $\mathcal{N}_{i,\text{sc}}$. Then by step (IV), all agents $i \in \mathcal{N}_{\text{ge}}$ would reset Υ_i and start a new round, since $\Upsilon_i[j] \geq 1$, $\forall\, j \in \mathcal{N}$ and $j \notin \mathcal{N}_{i,\text{sc}}$, i.e., all neighbors except those in $\mathcal{N}_{i,\text{sc}}$ have made a progress in plan execution.

The above procedure repeats itself until all agents in $\mathcal{N}_{\text{sc}}$ finish their plan execution and become passive. Then, it holds that $\mathcal{N}_{i,\text{sc}} = \mathcal{N}_{\text{sc}}, \forall\, i \in \mathcal{N}_{\text{ge}}$ and the results from Theorem 11 apply directly, meaning each agent in $\mathcal{N}_{\text{ge}}$ can satisfy its local task. Thus, all agents satisfy their local tasks while fulfilling the relative-distance constraints for all time. □

It is obvious that the above three protocols have different applicabilities. Thus, considering three cases separately is important such that the users can choose the suitable protocol.

D. Real-Time Discrete Plan Adaptation

In the aforementioned approaches, the discrete plan of each agent is synthesized only once initially from Section IV-A and executed according to the hybrid control scheme in real-time, regardless of the agents' actual trajectories. However, due to the relative-distance constraints, one agent's actual trajectory may be different from the planned one, i.e., it may detour to other agent's goal region as stated in Section IV-B. Thus, given the agent's updated position, its initial plan τ_i might be inefficient in terms of cost defined by (4). Thus, we propose a discrete plan adaptation algorithm that ensures that the updated plan always fulfills the task and has the minimal *suffix cost* for any agent i with a general LTL formula defined as follows:

$$\mathbf{cost}\left(\overline{\mathbf{p}}_{i,\mathrm{suf}}(T)\right) \triangleq \max_{(\pi_s,\pi_g)\in\vartheta} \{W_i(\pi_s,\pi_g)\} \tag{24}$$

where $\vartheta = \{(\pi_{i0},\pi_{\ell_1}),(\pi_{\ell_k},\pi_{\ell_{k+1}}),\forall\, \pi_{\ell_k},\pi_{\ell_{k+1}} \in \overline{\mathbf{p}}_{i,\mathrm{suf}}(T)\}$; and $\overline{\mathbf{p}}_{i,\mathrm{suf}}(T)$ is the suffix part of the effective path $\overline{\mathbf{p}}_i(T)$ that will be repeated infinitely often to satisfy a general LTL formula φ_i. Now, assume that at time $t_0 > 0$, agent i finishes executing its current plan suffix $\tau_{i,\mathrm{suf}}$ once. Denote by $\tau^-_{i,\mathrm{suf}}(t_0)$ the plan suffix before the plan update at time $t_0 > 0$ and $\tau^+_{i,\mathrm{suf}}(t_0)$ the plan suffix after the update. Denote by $\mathtt{word}^-_i(0,t_0)$ the past sequence of services provided by agent i during the time period $[0,t_0]$. Since the corresponding word suffix of $\tau^+_{i,\mathrm{suf}}(t_0)$ is given by $\tau^+_{i,\mathrm{suf}}(t_0)|_{\Sigma_i}$, the planned sequence of services by agent i during the time period $[t_0,T)$ is given by $\mathtt{word}^+_i(t_0,T) = \tau^+_{i,\mathrm{suf}}(t_0)|_{\Sigma_i}$, where $T=\infty$. Denote by $\mathtt{word}^\star_i(T)$ the complete word from $t=0$ to $t=T$, which is the complete sequence of services provided by agent i. Given the updated plan $\tau^+_i(t_0)$, $\mathtt{word}^\star_i(T)$ can be computed by concatenating the word during time $t=[0,t_0]$ and the word during time $t=[t_0,T)$ as follows:

$$\mathtt{word}^\star_i(T) = \mathtt{word}^-_i(0,t_0)\mathtt{word}^+_i(t_0,T). \tag{25}$$

On the other hand, $\tau^+_{i,\mathrm{suf}}(t_0)$ determines the suffix of an effective path after time t_0 by $\overline{\mathbf{p}}^+_{i,\mathrm{suf}}(t_0) = \tau^+_{i,\mathrm{suf}}(t_0)|_{\Pi_i}$, of which the suffix cost is given by (24).

Problem 2: Find an updated plan suffix $\tau^+_{i,\mathrm{suf}}(t_0)$ for agent i such that: (I) $\mathtt{word}^\star_i(T)$ by (25) satisfies φ_i; (II) the effective path suffix $\overline{\mathbf{p}}^+_{i,\mathrm{suf}}(t_0)$ has the minimal cost by (24).

The solution consists of two main steps: (I) we compute the set of all product states $Q'_{\mathcal{P},i,t_0} \subseteq Q_{\mathcal{P},i}$ that are reachable from the initial states $Q_{\mathcal{P},i,0}$ given the past effective path $\overline{\mathbf{p}}^-_i(0,t_0)$ and the past sequence of services $\mathtt{word}^-_i(0,t_0)$ provided by agent i during time $[0,t_0)$. In particular, it can be computed by iterating through the sequence of input word by $\mathtt{word}^-_i(0,t_0)$ and computes the set of successors recursively, while at the same time ensuring that it is compliant with the agent's past effective path by $\overline{\mathbf{p}}^-_i(0,t_0)$. Note that $Q'_{\mathcal{P},i,t_0}$ can be maintained by each agent along with the plan execution procedure described in Section IV-C3; (II) we compute firstly the intersection $F' = F_{\mathcal{P},i} \cap Q'_{\mathcal{P},i,t_0}$, which is always non-empty as $Q'_{\mathcal{P},i,t_0}$ contains at least one accepting state in $F_{\mathcal{P},i}$ as agent i has finished executing its plan suffix once at $t=t_0$. Then, the graph search algorithm described in Section IV-A is applied with slight modifications to compute the minimal-bottleneck prefix and cycle, where $Q_{\mathcal{P},i,0}$ is replaced by $Q'_{\mathcal{P},i,t_0}$ and $F_{\mathcal{P},i}$ is replaced by F'. Denote by $\varrho_{i,\mathrm{suf}}$ the minimal-bottleneck cycle from $v^\star_f$ back to itself, which is computed based on $P_{v^\star_f}$. Then, $\tau^+_{i,\mathrm{suf}}(t_0)$ is determined by the projection of $\varrho_{i,\mathrm{suf}}$ onto $\mathcal{T}_i$ and, thus, $\overline{\mathbf{p}}^+_{i,\mathrm{suf}}(t_0)$ is given as the projection of $\tau^+_{i,\mathrm{suf}}(t_0)$ onto Π_i. The worst-case computational complexity of the above synthesis algorithm can be determined in a similar way to the initial synthesis algorithm as $\mathcal{O}(|Q_{\mathcal{P},i}|^3 \cdot |F_{\mathcal{P},i}|)$, where $|Q_{\mathcal{P},i}|$ is the number of states in $\mathcal{P}_i$.

Lemma 13: $\tau^+_{i,\mathrm{suf}}(t_0)$ derived above solves Problem 2.

Proof: (Sketch) For part (I) of Problem 2: by how the reachable set $Q'_{\mathcal{P},i,t_0}$ is computed, we know that for any state $q'_s \in Q'_{\mathcal{P},i,t_0}$, there exists a path in $\mathcal{P}_i$ from one initial state $q'_0 \in Q_{\mathcal{P},i,0}$ to q'_s, which corresponds to $\mathtt{word}^-_i(0,t_0)$. Furthermore, $\tau^+_{i,\mathrm{suf}}(t_0)$ is generated by enforcing its word $\mathtt{word}^+_i(t_0,T)$ corresponds to a path in $\mathcal{P}_i$ which starts from one state $q'_f \in Q'_{\mathcal{P},i,t_0} \cap F'$ and cycles back to itself. By concatenating $\mathtt{word}^-_i(0,t_0)$ and $\mathtt{word}^+_i(t_0,T)$ as in (25), it is guaranteed that the complete word $\mathtt{word}^\star_i(T)$ corresponds to a path in $\mathcal{P}_i$ from the initial state q'_0 to an accepting state q'_f and then back to itself, which is an accepting path of $\mathcal{P}_i$ by definition. Regarding part (II): the synthesis algorithm ensures that the updated plan suffix $\tau^+_{i,\mathrm{suf}}(t_0)$ of $\mathcal{P}_i$ minimizes the cost by (24), which gives the effective path suffix $\overline{\mathbf{p}}^+_{i,\mathrm{suf}}(t_0)$ with the minimal suffix cost. This completes the proof. □

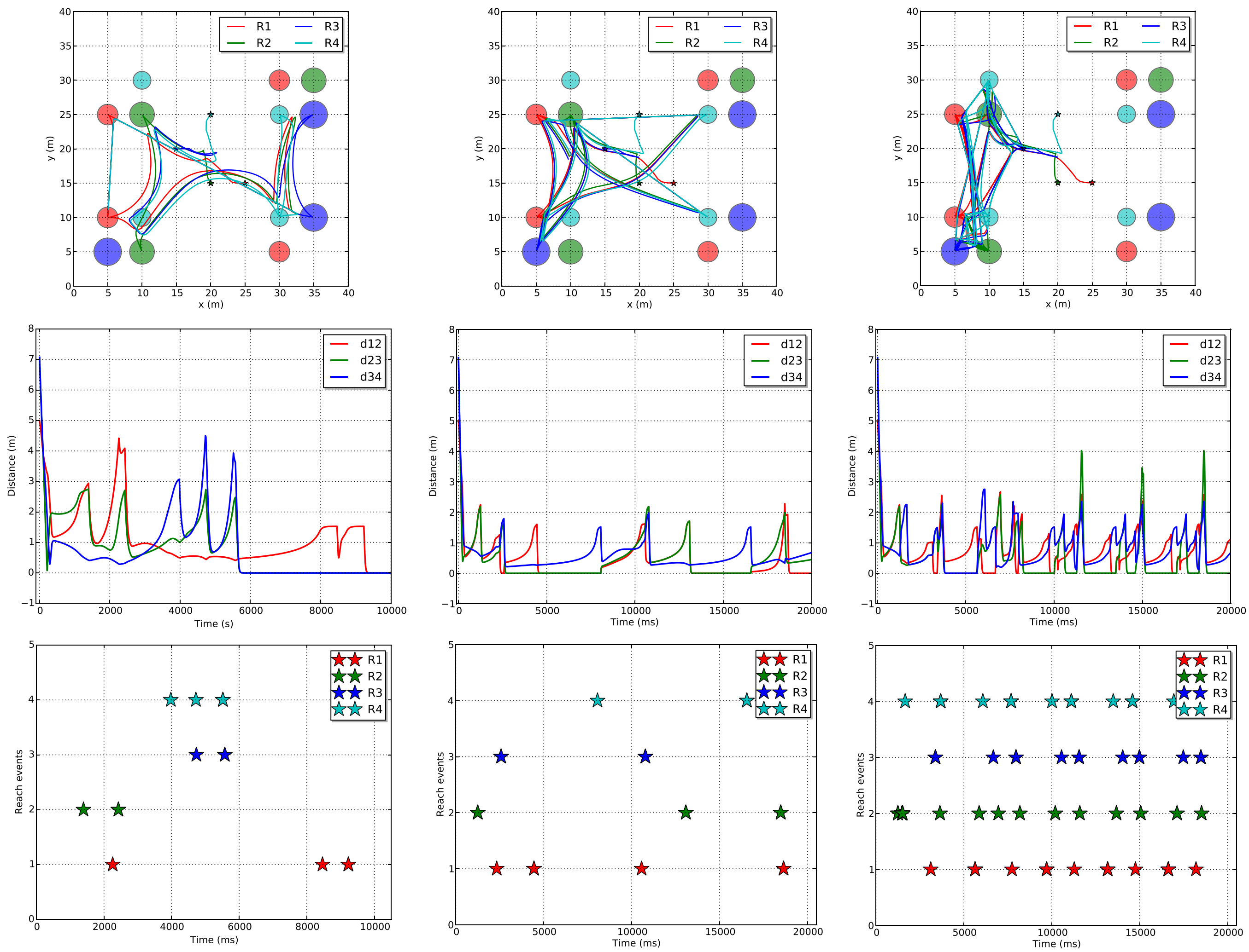
Now, we discuss how to integrate the above plan adaptation scheme with the switching policies described earlier. We consider here only the policy $\mathbf{P_{ge}}$ from Section IV-C2 and policy $\mathbf{P_{mx}}$ from Section IV-C3. We propose that *each* agent with a general LTL task specification updates its plan whenever it finishes executing its plan suffix *once*, by executing the adaptation algorithm above to compute the updated plan suffix $\tau^+_{i,\mathrm{suf}}$. To be more specific, for policy $\mathbf{P_{ge}}$, every agent follows the switching protocol and updates its plan suffix whenever it finishes executing its current plan suffix; for policy $\mathbf{P_{mx}}$, all agents follow the protocol but only the agents within $\mathcal{N}_{\mathrm{ge}}$ update its plan suffix whenever it finishes executing its current plan suffix. Then, the continuous controller and the switching protocol are updated accordingly given the updated plan suffix.

Lemma 14: For all agent $i \in \mathcal{N}$, the final execution word $\mathtt{word}_i(T)$ satisfies φ_i for $T=\infty$, after applying the plan adaptation scheme described above.

Proof: For policy $\mathbf{P_{ge}}$, since the plan adaptation is performed by every agent $i \in \mathcal{N}$ when it finishes executing its suffix once, Lemma 13 guarantees that $\tau^+_{i,\mathrm{suf}}(t_0)$ after the update at $t=t_0$ is a cyclic suffix containing an accepting state of $\mathcal{P}_i$. Thus, at least one accepting state of $\mathcal{P}_i$ is visited between two consecutive plan updates of agent i. Moreover, as shown in Lemma 10 and Theorem 12, any plan suffix has finite length and can be executed in finite time. The set of accepting states of $\mathcal{P}_i$ will be visited infinitely many times as $T=\infty$. Since the number of accepting states in $\mathcal{P}_i$ is finite, at least one of the accepting states will be visited infinitely often by $\mathtt{word}^\star_i(T)$ as $T=\infty$. Thus, by definition, $\mathtt{word}^\star_i(T)$ satisfies φ_i when $T=\infty, \forall\, i \in \mathcal{N}$. By policy $\mathbf{P_{mx}}$, any agent $i \in \mathcal{N}_{\mathrm{sc}}$ does not update its plan, the result from Theorem 12 still holds, while for any agent $i \in \mathcal{N}_{\mathrm{ge}}$, the analysis is similar to policy $\mathbf{P_{ge}}$. □

Theorem 15: When combining the plan adaption scheme with the protocol $\mathbf{P_{ge}}$ or $\mathbf{P_{mx}}$, it is guaranteed that $\forall\, i \in \mathcal{N}$, the local task φ_i is satisfied by $\mathbf{x}_i(T)$ and $\|x_i(t) - x_j(t)\| < r$, $\forall\,(i,j) \in E_0(0)$ and $\forall\, t > 0$, where $T=\infty$.

Proof: (Sketch) The first part regarding the task satisfaction is a direct extension of Lemma 14 above, while the second part regarding relative-distance constraints can be shown in a similar way as in Theorems 11 and 12. □



(a) (b) (c)

Fig. 1. (a) Top: agents' respective regions of interest in red, green, blue, and cyan, respectively, and their trajectories under policy $\mathbf{P_{sc}}$. All agents accomplish their sc-LTL tasks after 9 s. Middle: the evolution of pair-wise distances $\|x_{12}\|$, $\|x_{23}\|$, $\|x_{34}\|$, which all satisfy the distance constraints (below 7.5 m). Bottom: the time instants when the agents reach their goal regions and provide the set of planned services. (b) agents' trajectories under policy $\mathbf{P_{ge}}$ with general LTL tasks. (c) agents' trajectories under policy $\mathbf{P_{ge}}$ with general LTL tasks, after incorporating the plan adaptation algorithm. The bottom figure can be compared with Fig. 1(b).

V. Simulation

In the following case study, we simulate a team of four autonomous robots $\mathcal{N} = \{\mathfrak{R}_1, \ldots, \mathfrak{R}_4\}$ subject to the dynamics (1) in a bounded, obstacle-free workspace of 40×40 meters (m). Each robot $\mathfrak{R}_i$ is given a local task specified as sc-LTL or LTL formulas φ_i. All algorithms and modules were implemented in Python 2.7. Simulations were carried out on a desktop computer (3.06 GHz Duo CPU and 8GB of RAM) with a simulation stepsize set to 1 ms.

As shown in Fig. 1(a), several regions of interest for each agent are placed in top-left, top-right, bottom-right and bottom-left corners of the workspace and they all satisfy Assumption 1 with $c_{\max} = 40$ and $r_{min} = 2$. Particularly, we consider the following aspects:

(1) Regions of interest. Agent $\mathfrak{R}_1$ is an aerial vehicle with four regions of interest, denoted by $\Pi_1 = \{\pi_{1\text{tl}}, \pi_{1\text{tr}}, \pi_{1\text{br}}, \pi_{1\text{bl}}\}$ shown in red; agent $\mathfrak{R}_2$ is a ground vehicle with three regions of interest $\Pi_2 = \{\pi_{2\text{tl}}, \pi_{2\text{tr}}, \pi_{2\text{bl}}\}$ shown in green; agent $\mathfrak{R}_3$ is also a ground vehicle with $\Pi_3 = \{\pi_{3\text{tr}}, \pi_{3\text{br}}, \pi_{3\text{bl}}\}$ shown in blue; agent $\mathfrak{R}_4$ is an aerial vehicle with $\Pi_4 = \{\pi_{4\text{tl}}, \pi_{4\text{tr}}, \pi_{4\text{br}}, \pi_{4\text{bl}}\}$ shown in cyan. Note that we only consider the planar position of all agents.

(2) Services. Agent $\mathfrak{R}_1$ is capable of providing two kinds of services, i.e., surveillance over an area (denoted by σ_{11}) and assistance for ground operations (denoted by σ_{12}). Thus, its set of atomic propositions is given by $\Sigma_1 = \{\sigma_{11}, \sigma_{12}\}$. Agent $\mathfrak{R}_4$ can provide the analogous kind of services as $\mathfrak{R}_1$, denoted by $\Sigma_4 = \{\sigma_{41}, \sigma_{42}\}$. Moreover, agent $\mathfrak{R}_2$ is capable of providing three kinds of services, i.e., food delivery (denoted by σ_{21}), water delivery (denoted by σ_{22}), and transportation (denoted by σ_{23}). Thus, its set of atomic propositions is $\Sigma_2 = \{\sigma_{21}, \sigma_{22}, \sigma_{23}\}$. Agent $\mathfrak{R}_3$ can provide the analogous kind of services as $\mathfrak{R}_2$, denoted by $\Sigma_3 = \{\sigma_{31}, \sigma_{32}, \sigma_{33}\}$.

(3) Region labeling. The aerial assistance service is available at two regions of interest of $\mathfrak{R}_1$ while the surveillance service is available at the other two regions. Namely, $L_1(\pi_{1\text{tl}}) = L_1(\pi_{1\text{br}}) = \{\sigma_{11}\}$ and $L_1(\pi_{1\text{tr}}) = L_1(\pi_{1\text{bl}}) = \{\sigma_{12}\}$. Similar statements hold for agent $\mathfrak{R}_4$, i.e., $L_4(\pi_{4\text{tl}}) = L_4(\pi_{4\text{tr}}) = \{\sigma_{41}\}$, $L_4(\pi_{4\text{bl}}) = L_4(\pi_{4\text{br}}) = \{\sigma_{42}\}$. While for agent $\mathfrak{R}_2$, the food delivery, water delivery, and transportation services are available at its regions of interest, respectively. Namely, $L_2(\pi_{2\text{tl}}) = \{\sigma_{21}\}$, $L_2(\sigma_{2\text{tr}}) = \{\sigma_{22}\}$, $L_2(\pi_{2\text{bl}}) = \{\sigma_{23}\}$. Similar statements hold for agent $\mathfrak{R}_3$, i.e., $L_3(\pi_{3\text{tr}}) = \{\sigma_{31}\}$, $L_3(\pi_{3\text{br}}) = \{\sigma_{32}\}$, $L_3(\pi_{3\text{bl}}) = \{\sigma_{33}\}$.

(4) Network graph. The agents have a uniform neighboring radius as $r = 8$ m and the design parameter needed in Definition 2 is $\kappa = 0.5$ m. They start from [25, 15], [20, 15], [15, 20] and [20, 25] in the 2D workspace. Thus, the initial edge set of $G(0)$ is given by $E_0(0) = \{(\mathfrak{R}_1, \mathfrak{R}_2), (\mathfrak{R}_2, \mathfrak{R}_3), (\mathfrak{R}_3, \mathfrak{R}_4)\}$. The upper bound by (23) is $\varepsilon < \varepsilon_{\min} \approx 0.031$ and we choose $\varepsilon = 0.03$.

We consider two cases of the agent task specifications: one with sc-LTL formulas and one with general LTL formulas.

(I) *sc-LTL Task Specifications*. The finite-time local task for agent $\mathfrak{R}_1$ or $\mathfrak{R}_4$ is to first provide the surveillance, assistance service to the ground vehicles and then another surveillance service in this sequence to regions required. Namely, $\varphi_1^s = \diamond(\sigma_{12} \wedge \diamond(\sigma_{11} \wedge \diamond\sigma_{12}))$ and $\varphi_4^s = \diamond(\sigma_{42} \wedge \diamond(\sigma_{41} \wedge \diamond\sigma_{42}))$. On the other hand, the finite-time local task for agent $\mathfrak{R}_2$ or $\mathfrak{R}_3$ is to first deliver food or water and then provide transportation service, which is formalized as $\varphi_2^s = \diamond(\sigma_{21} \vee \sigma_{22}) \wedge \diamond\sigma_{23}$ and $\varphi_3^s = \diamond(\sigma_{31} \vee \sigma_{32}) \wedge \diamond\sigma_{33}$.

The synthesized discrete plans derived by the algorithm described in Section IV-A are as follows: agent $\mathfrak{R}_1$ needs to provide surveillance at region $\pi_{1\text{bl}}$, assistance at region $\pi_{1\text{tl}}$ and then surveillance at region $\pi_{1\text{bl}}$, i.e., $\tau_1 = (\pi_{1\text{bl}}, \{\sigma_{12}\})(\pi_{1\text{tl}}, \{\sigma_{11}\})(\pi_{1\text{bl}}, \{\sigma_{12}\})$; agent $\mathfrak{R}_2$ would supply food at region $\pi_{2\text{tl}}$ and transportation at region $\pi_{2\text{bl}}$ while agent $\mathfrak{R}_3$ would supply food at region $\pi_{3\text{tl}}$ and transportation at region $\pi_{3\text{bl}}$. Namely, $\tau_2 = (\pi_{2\text{tl}}, \{\sigma_{21}\})(\pi_{2\text{bl}}, \{\sigma_{23}\})$ and $\tau_3 = (\pi_{3\text{tr}}, \{\sigma_{31}\})(\pi_{3\text{br}}, \{\sigma_{33}\})$. At last, agent $\mathfrak{R}_4$ needs to provide surveillance at region $\pi_{4\text{bl}}$, assistance at region $\pi_{4\text{tl}}$ and then surveillance at region $\pi_{4\text{bl}}$, i.e., $\tau_4 = (\pi_{4\text{br}}, \{\sigma_{41}\})(\pi_{4\text{tr}}, \{\sigma_{42}\})(\pi_{4\text{tl}}, \{\sigma_{41}\})$. It can be verified that they all satisfy the respective local tasks. At $t = 0$, the switching policy $\mathbf{P_{sc}}$ from Section IV-C1 is applied. It takes around 9 s for all agents to accomplish the execution of their local plans. The complete agent trajectories are shown in Fig. 1(a), where the distances between the neighboring agents along with times of reaching the agents' respective progressive goal regions are also plotted. In addition, the time instants when each agent reaches their goal regions are shown to illustrate the progressive plan execution.

(II) *General LTL task specifications*. In this case, all agents' local tasks are specified as general LTL formulas over the services. The task of agent $\mathfrak{R}_1$ is to periodically provide both the surveillance and assistance services σ_{11} and σ_{12} at the required regions, which is represented by $\phi_1 = \Box\diamond\sigma_{11} \wedge \Box\diamond\sigma_{12}$; the task of agents $\mathfrak{R}_2$ and $\mathfrak{R}_3$ are similar, which is to periodically provide either food, water supply, or transportation service at desired regions, which is formalized as $\phi_2 = \Box\diamond(\sigma_{21} \vee \sigma_{22} \vee \sigma_{23})$ and $\phi_3 = \Box\diamond(\sigma_{31} \vee \sigma_{32} \vee \sigma_{33})$; at last, the task of agent $\mathfrak{R}_4$ is to periodically provide both the surveillance and assistance services at the required regions, which is represented by $\phi_4 = \Box\diamond\sigma_{41} \wedge \Box\diamond\sigma_{42}$.

The synthesized discrete plans derived by the algorithm described in Section IV-A are as follows: agent $\mathfrak{R}_1$ would provide assistance at region $\pi_{1\text{bl}}$ and then surveillance at region $\pi_{1\text{tl}}$, which is repeated infinitely often, i.e., $\tau_1 = ((\pi_{1\text{bl}}, \{\sigma_{12}\})((\pi_{1\text{tl}}, \{\sigma_{11}\}))^\omega$; agent $\mathfrak{R}_2$ would supply food at region $\pi_{2\text{tl}}$, repetitively, and agent $\mathfrak{R}_3$ would provide transportation service at region $\pi_{3\text{bl}}$, repetitively. Namely, $\tau_2 = (\pi_{2\text{tl}}, \{\sigma_{21}\})^\omega$ and $\tau_3 = (\pi_{3\text{bl}}, \{\sigma_{33}\})^\omega$; at last, agent $\mathfrak{R}_4$ would provide surveillance at region $\pi_{4\text{br}}$ and then assistance at region $\pi_{4\text{tr}}$, which is repeated infinitely often, i.e., $\tau_4 = ((\pi_{4\text{br}}, \{\sigma_{41}\})(\pi_{4\text{tr}}, \{\sigma_{42}\}))^\omega$.

The simulation results for the activity switching protocol $\mathbf{P_{ge}}$ from Section IV-C2 are illustrated in Fig. 1(b). The functions f_{prob} and f_{cond} were chosen in a way that allows to partially trade fairness of activity switching for increased efficiency of plan executions measured in terms of the distance traveled between consecutive visits to progressive goal regions. More specifically, an agent is not switched to passive immediately after it reaches one of its goal region. Rather than that, it has the following probability of remaining active: $\mathbf{Pr}(b_i = 1) = e^{-\alpha_i \Upsilon_i[i](t-\chi_i)}$ if $\Upsilon_i[i] \cdot (t - \chi_i) < \bar{\chi}_i$; and $\mathbf{Pr}(b_i = 1) = 0$ if $\Upsilon_i[i] \cdot (t - \chi_i) \geq \bar{\chi}_i$, where $\bar{\chi}_i = 5$ and $\alpha_i = 1$. The probability of remaining active decreases with the increasing time elapsed since the current round started and with the increasing number agent $\mathfrak{R}_i$'s own progressive goal region was visited. Note that there exists a finite $T \in (T_{\circlearrowleft_{m-1}}, T_{\circlearrowleft_m})$, such that $\Upsilon_i[i] \cdot (t - \chi_i) \geq \bar{\chi}_i$ for all $t \in [T, T_{\circlearrowleft_m}]$, hence each agent $\mathfrak{R}_i$ is guaranteed to be switched to passive control mode eventually. The selected function does not necessarily yield a monotonic decrease of the total number of active agents in the team and is useful when one agent has close goal regions.

At last, to demonstrate the effectiveness of the local plan adaptation technique proposed in Section IV-D, we combine the protocol $\mathbf{P_{ge}}$ and the real-time adaptation algorithm. The results are shown in Fig. 1(c), which is significantly different from the results in Fig. 1(b): Agent $\mathfrak{R}_4$ adapts its plan to visit $\pi_{4\text{tl}}$ with service σ_{42} while agent $\mathfrak{R}_1$ is reaching $\pi_{1\text{tl}}$ with service σ_{11}. Then agent $\mathfrak{R}_4$ adapts its plan to visit region $\pi_{4\text{bl}}$ and provide the surveillance service there and agent $\mathfrak{R}_2$ adapts its plan to visit region $\pi_{2\text{bl}}$ and supply water there while agent $\mathfrak{R}_1$ is reaching the region $\pi_{1\text{bl}}$. Consequently, the agents reach their goal regions much more often than before when the real-time adaptation algorithm is not applied, which can be confirmed by comparing from the time instants when each agent reaches its goal region in Fig. 1(b) and (c), respectively.

VI. Conclusion and Future Work

We proposed a distributed communication-free control scheme for multi-agent systems to fulfill locally-assigned tasks as general or sc-LTL formulas, while subject to relative-distance constraints. Future work include handling collision avoidance among the agents, uncertainties in the relative-state measurements and more complex agent dynamics.

REFERENCES

[1] C. Baier and J.-P. Katoen, *Principles of Model Checking*. Cambridge, MA, USA: MIT Press, 2008.

[2] Y. Chen, X. C. Ding, A. Stefanescu, and C. Belta, "Formal approach to the deployment of distributed robotic teams," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 158–171, 2012.

[3] D. V. Dimarogonas, S. G. Loizou, K. J. Kyriakopoulos, and M. M. Zavlanos, "A feedback stabilization and collision avoidance scheme for multiple independent non-point agents," *Automatica*, vol. 42, no. 2, pp. 229–243, 2006.

[4] G. E. Fainekos, S. G. Loizou, and G. J. Pappas, "Translating temporal logic to controller specifications," in *Proc. IEEE Conf. Decision Control (CDC)*, 2006, pp. 899–904.

[5] I. Filippidis, D. V. Dimarogonas, and K. J. Kyriakopoulos, "Decentralized multi-agent control from local LTL specifications," in *Proc. IEEE Conf. Decision Control (CDC)*, 2012, pp. 6235–6240.

[6] M. Franceschelli, M. B. Egerstedt, and A. Giua, "Motion probes for fault detection and recovery in networked control systems," in *Proc. Amer. Control Conf. (ACC)*, 2008, pp. 4358–4363.

[7] P. Gastin and D. Oddoux, "Fast LTL to Büchi automata translation," in *Proc. Int. Conf. Comput. Aided Verification (CAV)*, 2001, pp. 53–65.

[8] C. Godsil and G. Royle, *Algebraic Graph Theory*. Berlin, Germany: Springer Graduate Texts in Math., 2001.

[9] M. Guo and D. V. Dimarogonas, "Multi-agent plan reconfiguration under local LTL specifications," *Int. J. Robot. Res.*, vol. 34, no. 2, pp. 218–235, 2015.

[10] M. Guo, J. Tumova, and D. V. Dimarogonas, "Cooperative decentralized multi-agent control under local LTL tasks and connectivity constraints," in *Proc. IEEE Conf. Decision Control (CDC)*, 2014, pp. 75–80.

[11] M. Guo, J. Tumova, and D. V. Dimarogonas, "Hybrid control of multi-agent systems under local temporal tasks and relative-distance constraints," in *Proc. IEEE Conf. Decision Control (CDC)*, 2015, pp. 1701–1706.

[12] M. Guo, M. M. Zavlanos, and D. V. Dimarogonas, "Controlling the relative agent motion in multi-agent formation stabilization," *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 820–826, 2014.

[13] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, MA, USA: Cambridge Univ. Press, 2012.

[14] M. Ji and M. B. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Trans. Robot.*, vol. 23, no. 4, pp. 693–703, 2007.

[15] S. Karaman and E. Frazzoli, "Vehicle routing with temporal logic specifications: Applications to multi-UAV mission planning," *Int. J. Robust Nonlinear Control*, vol. 21, pp. 1372–1395, 2011.

[16] H. K. Khalil and J. W. Grizzle, *Nonlinear Systems*. Upper Saddle River, NJ, USA: Prentice Hall, 2002.

[17] M. Kloetzer, X. C. Ding, and C. Belta, "Multi-robot deployment from LTL specifications with reduced communication," in *Proc. IEEE Conf. Decision Control Europ. Control Conf. (CDC-ECC)*, 2011, pp. 4867–4872.

[18] P. Kondaxakis, J. Pajarinen, and V. Kyrki, "Real-time recognition of pointing gestures for robot to robot interaction," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Syst.*, 2014, pp. 2621–2626.

[19] O. Kupferman and M. Y. Vardi, "Model checking of safety properties," *Formal Methods Syst. Design*, vol. 19, pp. 291–314, 2001.

[20] S. M. LaValle, *Planning Algorithms*. Cambridge, MA, USA: Cambridge University Press, 2006.

[21] S. G. Loizou and K. J. Kyriakopoulos, "Automated planning of motion tasks for multi-robot systems," in *Proc. IEEE Conf. Decision Control (CDC)*, 2005, pp. 78–83.

[22] S. G. Loizou and K. J. Kyriakopoulos, "Automatic synthesis of multiagent motion tasks based on ltl specifications," in *Proc. IEEE Conf. Decision Control (CDC)*, 2004, pp. 153–158.

[23] J. Lygeros, F. Ramponi, and C. Wiltsche, "Synthesis of an asynchronous communication protocol for search and rescue robots," in *Proc. Europ. Control Conf. (ECC)*, 2013, pp. 1256–1261.

[24] P. Ogren, M. Egerstedt, and X. Hu, "A control Lyapunov function approach to multi-agent coordination," in *Proc. IEEE Conf. Decision Control (CDC)*, 2001, pp. 1150–1155.

[25] M. Mazo and P. Tabuada, "Decentralized event-triggered control over wireless sensor/actuator networks," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2456–2461, 2011.

[26] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proc. Amer. Control Conf. (ACC)*, 2005, pp. 1859–1864.

[27] E. Rimon and D. E. Koditschek, "Exact robot navigation using cost functions: The case of distinct spherical boundaries in E^n," in *Proc. IEEE Int. Conf. Robotics Autom. (ICRA)*, 1988, pp. 1791–1796.

[28] I. Saha, R. Ramaithitima, V. Kumar, G. Pappas, and S. Seshia, "Automated composition of motion primitives for multi-robot systems from safe LTL specifications," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2014, pp. 1525–1532.

[29] J. Tumova and D. V. Dimarogonas, "A receding horizon approach to multi-agent planning from LTL specifications," in *Proc. Amer. Control Conf. (ACC)*, 2014, pp. 1775–1780.

[30] A. Ulusoy, S. L. Smith, X. C. Ding, C. Belta, and D. Rus, "Optimality and robustness in multi-robot path planning with temporal logic constraints," *Int. J. Robot. Res.*, vol. 32, no. 8, pp. 889–911, 2013.

[31] M. M. Zavlanos, M. B. Egerstedt, and G. J. Pappas, "Graph-theoretic connectivity control of mobile robot networks," *Proc. IEEE*, vol. 99, no. 9, pp. 1525–1540, 2011.

[32] M. M. Zavlanos, A. Jadbabaie, and G. J. Pappas, "Flocking while preserving network connectivity," in *Proc. IEEE Conf. Decision Control (CDC)*, 2007, pp. 2919–2924.

Meng Guo (S'14) received the M.Sc. degree in system, control and robotics in 2011 from the KTH Royal Institute of Technology, Sweden. Currently, he is a Ph.D. student at the Department of Automatic Control, School of Electrical Engineering, KTH. His main research interest includes distributed motion and task planning of multi-agent systems and formal control synthesis.

Jana Tumova (M'14) received her Ph.D. degree in computer science from Masaryk University in Brno, Czech Republic in 2013. She is currently a postdoctoral researcher at the Automatic Control Department at KTH Royal Institute of Technology in Stockholm, Sweden. Her research interests include formal verification, control synthesis, and temporal logics in general, as well as formal methods applied in robot motion planning, control and analysis of dynamical and hybrid systems and multi-agent control.

Dimos V. Dimarogonas (M'10) was born in Athens, Greece, in 1978. He received the Diploma in electrical and computer engineering in 2001 and the Ph.D. in mechanical engineering in 2007, both from the National Technical University of Athens (NTUA), Greece. Between May 2007 and February 2009, he was a Postdoctoral Researcher at the Automatic Control Laboratory, School of Electrical Engineering, ACCESS Linnaeus Center, KTH Royal Institute of Technology, Stockholm, Sweden. Between February 2009 and March 2010, he was a Postdoctoral Associate at the Laboratory for Information and Decision Systems (LIDS) at the Massachusetts Institute of Technology (MIT), Boston, MA, USA. He is currently an Associate Professor at the Automatic Control Laboratory, ACCESS Linnaeus Center, KTH Royal Institute of Technology, Stockholm, Sweden. His current research interests include Multi-Agent Systems, Hybrid Systems and Control, Robot Navigation and Networked Control. He was awarded a Docent in Automatic Control from KTH in 2012. He serves in the Editorial Board of Automatica, the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING and the IET Control Theory and Applications and is a member of IEEE and the Technical Chamber of Greece. He received an ERC Starting Grant from the European Commission for the proposal BUCOPHSYS in 2014 and was awarded a Wallenberg Academy Fellow grant in 2015.