2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Decentralized Goal Assignment and Trajectory Generation in Multi-Robot Networks: A Multiple Lyapunov Functions Approach

Dimitra Panagou, Matthew Turpin and Vijay Kumar

Abstract— **This paper considers the problem of decentralized goal assignment and trajectory generation for multi-robot networks when only local communication is available, and proposes an approach based on methods related to switched systems and set invariance. A family of Lyapunov-like functions is employed to encode the (local) decision making among candidate goal assignments, under which the agents pick the assignment which results in the shortest total distance to the goals. An additional family of Lyapunov-like barrier functions is activated in the case when the optimal assignment may lead to colliding trajectories, thus maintaining system safety while preserving the convergence guarantees. The proposed switching strategies give rise to feedback control policies which are scalable as the number of agents increases, and therefore are suitable for applications including first-response deployment of robotic networks under limited information sharing. Simulations demonstrate the efficacy of the proposed method.**

I. Introduction

Task (target) assignment problems in multi-agent systems have received great interest within the robotics and controls communities in the past couple of years, in part because they encode the accomplishment of various objectives in, among others, surveillance, exploration and coverage applications.

A common thread in such problems is the development of algorithms which assign targets to agents by optimizing a predefined criterion and while meeting certain performance guarantees. The interest of researchers often focuses on the optimization aspects of the problem and the associated computational complexity under (quite) relaxed assumptions, which may not be acceptable in realistic settings. For instance, first-response or search-and-rescue missions using robotic agents (such as aerial robots) typically involve multiple tasks that, on one hand, may need to be performed as quickly as possible, yet on the other hand they typically can be accomplished only when information sharing is available, and under tight safety guarantees. In the interest of space, here we can not provide an overview of relevant formulations and approaches on target assignment problems in multi-robot networks. A more detailed introduction and literature review is provided in our previous work [1]; the reader is also referred to [2]–[7] and the references therein.

Dimitra Panagou is with the Coordinated Science Laboratory, College of Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA dpanagou@illinois.edu

Matthew Turpin and Vijay Kumar are with the GRASP Lab, School of Engineering and Applied Science, University of Pennsylvania, Philadelphia, PA, USA (mturpin, kumar)@seas.upenn.edu

We gratefully acknowledge the support of ARL grant W911NF-08-2-0004, ONR grant N00014-09-1-1051, and TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

This paper builds upon our previous work [1] and proposes algorithms which concurrently address the problems of: (P_1) Assigning goal locations (targets) to agents by minimizing a cost function, defined as the total distance to the goals. (P_2) Designing *feedback* control policies which guarantee: (i) the convergence of the agents to their assigned goals, (ii) that the resulting trajectories are collision-free. The key specifications in the proposed formulation are: (S_1) Agents and goal locations are interchangeable, which means that the mission is considered accomplished when each agent has converged to some goal location. (S_2) Information exchange between a pair of agents is reliable only when they lie within a certain communication range, which means that the decision making on the optimal goal assignment can be performed only locally, i.e., in a decentralized fashion. (S_3) Agents are modeled as non-point robots, which not only means that avoiding collisions is a non-negligible objective, but is of top priority even in the expense of resorting to suboptimal paths, if necessary, to the goals.

To this end, we formulate the goal assignment and trajectory generation problems into a control theoretic framework which is related to switched systems theory [8]. More specifically, we build our approach based on ideas and tools that rely on multiple Lyapunov-like functions [9]. We first encode the decision making on the optimal goal assignment (which in the sequel we call the Optimal Goal Assignment (OGA) policy) as a state-dependent switching logic among a family of candidate Lyapunov-like functions. Each Lyapunov-like function encodes the cost-to-go under a candidate goal assignment, that is, the sum of distances to the goals. The switching logic dictates that, when (a subset of) agents become(s) connected at some time instant t, they decide to switch to the Lyapunov-like function of minimum value at time t. We show that this decision making gives rise to a Globally Asymptotically Stable (GAS) switched system which furthermore does not suffer from Zeno trajectories. Then, based on our recent work in [10], we build an additional state-dependent switching logic which employs a family of Lyapunov-like barrier functions encoding both inter-agent collision avoidance and convergence to the goal locations determined by the OGA policy. This control policy (in the sequel called the Last Resort (LR) policy) provides sufficient conditions on determining whether the OGA policy is safe, and furthermore serves as a supervisor that takes action only when safety under the OGA policy is in stake. We show that the switching between the OGA policy and the LR policy results in asymptotically stable and safe trajectories for the multi-robot system, in the expense of possibly resorting to suboptimal paths; this situation appears only in the cases when the LR policy forces the agents to deviate from their optimal paths to the goals, in order to maintain system safety.

The paper is organized as follows: Section II gives the mathematical formulation of the problem. The proposed goal assignment and trajectory generation policies, along with the mathematical proofs which verify their correctness are given in Sections III and IV. Simulation results to evaluate their efficacy are included in Section V, while Section VI summarizes our conclusions and thoughts on future research.[1]

II. Problem Formulation

Assume N agents i and equal number of goal locations G_i, $i \in \mathcal{N} = \{1, 2, \ldots, \mathrm{N}\}$. The motion of each agent i is governed by single integrator dynamics:

$$\dot{\boldsymbol{r}}_i = \boldsymbol{u}_i, \tag{1}$$

where $\boldsymbol{r}_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^T$ is the position vector of agent i with respect to (w.r.t.) a global cartesian coordinate frame $\mathcal{G}$, and $\boldsymbol{u}_i$ is its control vector comprising the velocities u_{xi}, u_{yi} w.r.t. the frame $\mathcal{G}$. We assume that each agent i: (i) has access to its position $\boldsymbol{r}_i$ and velocities $\boldsymbol{u}_i$ via onboard sensors, (ii) can reliably exchange information with any agent $j \neq i$ which lies within its communication region $\mathcal{C}_i : \{\boldsymbol{r}_i \in \mathbb{R}^2, \boldsymbol{r}_j \in \mathbb{R}^2 \mid \|\boldsymbol{r}_i - \boldsymbol{r}_j\| \leq R_c\}$, where R_c is the communication range. In other words, a pair of agents (i, j) is connected as long as the distance $d_{ij} = \|\boldsymbol{r}_i - \boldsymbol{r}_j\| \leq R_c$.

It should be noted that here we assume that all robots have an initial goal assignment. This limits the approach to requiring at least as many robots as goal locations where some robots can instead have the command to remain stationary. While it may be possible to utilize virtual agents to allow more goals than robots, this is beyond the scope of this paper.

The task is considered completed as long as each agent has converged to some goal location, i.e., that the goals are interchangeable for each agent. This specification defines N! possible goal assignments $k \in \{1, 2, \ldots, \mathrm{N}!\}$.

Lemma 1: The position trajectories $\boldsymbol{r}_i(t)$ of agent i are Globally Exponentially Stable (GES) w.r.t. the k-th goal assignment under the control law:

$$\boldsymbol{u}_i^{(k)} = -\lambda_i \left(\boldsymbol{r}_i - \boldsymbol{r}_{g_i}^{(k)} \right), \tag{2}$$

where $i \in \{1, \ldots, \mathrm{N}\}$, $\lambda_i > 0$. The proof is trivial, see [11].

III. The Optimal Goal Assignment Policy

For simplicity, let us initially consider $\mathrm{N} = 2$ agents i, j which need to move to goal locations G_1, G_2, and let us build a switching logic realizing the local decision making on what we call the OGA.

Assume that the agents initiate at $t_0 = 0$ so that $d_{ij}(t_0) \leq R_c$, or that at some time instant $t_d > 0$ they lie within distance $d_{ij}(t_d) \leq R_c$ under some goal assignment $k \in \{1, 2\}$. We say that the agents are involved in a meeting event and a decision regarding on the goal assignment has to be made. The agents:

1) Exchange information on their current positions $\boldsymbol{r}_i(t_d)$, $\boldsymbol{r}_j(t_d)$ and goal locations $\boldsymbol{r}_{g_i}(t_d)$, $\boldsymbol{r}_{g_j}(t_d)$.
2) Compare the cost-to-go through the values of the Lyapunov functions:

$$V^{(k)}(\boldsymbol{r}(t_d)) = V_i^{(k)}(\boldsymbol{r}_i, \boldsymbol{r}_{g_i})(t_d) + V_j^{(k)}(\boldsymbol{r}_j, \boldsymbol{r}_{g_j})(t_d),$$

for all possible assignments $k \in \{1, 2\}$, where:

$$V_o^{(k)} = \|\boldsymbol{r}_o - \boldsymbol{r}_{g_o}\|, \;\; o \in \{i, j\}.$$

3) Move under: $V(\boldsymbol{r}(t_d)) = \min\left\{V^{(k)}(\boldsymbol{r}(t_d))\right\}$.

We refer to the decision making based on the logic described above as to the OGA policy. In the sequel we denote $V(\boldsymbol{r}(t_d))$ with $V(t_d)$, to keep the notation compact.

For $\mathrm{N} > 2$ agents the decision making on the optimal goal assignment involves the $\mathrm{N}_c \leq \mathrm{N}$ connected agents at time t_d. The agents exchange information on their positions and goal locations at time t_d, compare the cost-to-go through the Lyapunov functions $V^{(k)}$, $k \in \{1, 2, \ldots, \mathrm{N}_c!\}$, and determine the optimal assignment of robots to goals. The $\mathrm{N}_c!$ combinations of robots to goals result in intractable enumeration for all but the smallest problems but fortunately, there exists the Hungarian Algorithm [12], a $\mathcal{O}({N_c}^3)$ algorithm from the Operations Research community which optimally solves this problem in a centralized manner. Recent extensions [13] modify this approach for distributed systems.

Problem 1: We would like to establish that the OGA policy renders stable trajectories for the multi-robot system, i.e., that each agent does converge to some goal location.

A. Stability Analysis on the Switched Multi-robot System

To this end, we resort to control analysis tools for switched systems. The closed-loop dynamics of the i-th agent read:

$$\dot{\boldsymbol{r}}_i(t) = \boldsymbol{u}_i^{(k)} = -\lambda_i \left(\boldsymbol{r}_i(t) - \boldsymbol{r}_{g_i}^{(k)} \right). \tag{3}$$

The OGA policy gives rise to switched dynamics for each agent, in the sense that it may move under any of the N! possible goal assignments $k \in \{1, 2, \ldots, \mathrm{N}!\}$.

Denote $\boldsymbol{r} = \begin{bmatrix} {\boldsymbol{r}_1}^T & \ldots & {\boldsymbol{r}_\mathrm{N}}^T \end{bmatrix}^T$ the state vector of the multi-robot system, governed by the switched dynamics:

$$\dot{\boldsymbol{r}}(t) = \boldsymbol{f}_k(\boldsymbol{r}(t)), \tag{4}$$

$$\text{where:} \quad \boldsymbol{f}_k = \begin{bmatrix} \boldsymbol{u}_1^{(k)} \\ \vdots \\ \boldsymbol{u}_\mathrm{N}^{(k)} \end{bmatrix}, \; k \in \mathcal{K} = \{1, \ldots, \mathrm{N}!\}.$$

Let us consider the sequence of switching times $T = \{t_0, t_1, t_2, t_3, \ldots, t_n, \ldots\}$ and the switching sequence: $\Sigma = \{\boldsymbol{r}_0; (k_0, t_0), (k_1, t_1), \ldots, (k_n, t_n), \ldots\}$, $k_n \in \mathcal{K}$, $n \in \mathbb{N}$.

Theorem 1: The trajectories $\boldsymbol{r}(t)$ of the switched multi-robot system (4) are GAS w.r.t. the goal assignment k.

Proof: We consider the candidate Lyapunov-like functions $V^{(k)}$, $k \in \{1, \ldots, \mathrm{N}!\}$, encoding the motion of the

[1] In the interest of space, an extended version of this paper including also an overview of the theoretical tools from switched systems theory which are used throughout our analysis is available in [11].

agents under goal assignment k. Out of Lemma 1, each individual k-th subsystem, i.e. the motion of the multi-robot system under the k-th assignment, is GES. This implies that each $V^{(k)}$ is decreasing on the time intervals that the k-th subsystem is active. Furthermore, the OGA policy dictates that at the switching times $\{t_0, t_1, t_2, t_3, \dots,\}$ one has:

$$V^{(k_0)}(t_0) > V^{(k_1)}(t_1) > V^{(k_2)}(t_2) > V^{(k_3)}(t_3) > \dots,$$

i.e., the value of each Lyapunov-like function $V^{(k)}$ at the beginning of the time intervals when the k-th subsystem becomes active satisfies the decreasing condition, i.e., the switched system is Lyapunov stable.

To draw conclusions on the asymptotic stability, consider any pair of switching times $t_{k_1} < t_{k_2}$ when the k-th subsystem becomes active, the corresponding time intervals $[t_{k_1}, t_{k_1+1})$, $[t_{k_2}, t_{k_2+1})$ and note that:

$$V^{(k)}(t_{k_1}) > V^{(k)}(t_{k_1+1}), \quad \text{since the subsystem } k \text{ is active on } [t_{k_1}, t_{k_1+1}) \text{ and GES}$$

$$V^{(k)}(t_{k_1+1}) \geq V^{(l)}(t_{k_1+1}), \quad \text{out of the OGA Policy, } l \neq k$$

$$V^{(l)}(t_{k_1+1}) \geq V^{(k)}(t_{k_2}), \quad \text{out of the OGA Policy}$$

$$V^{(k)}(t_{k_2}) > V^{(k)}(t_{k_2+1}), \quad \text{since the subsystem } k \text{ is active on } [t_{k_2}, t_{k_2+1}) \text{ and GES}$$

which imply that: $V^{(k)}(t_{k_2+1}) < V^{(k)}(t_{k_1+1})$, i.e., that:

$$V^{(k)}(t_{k_2+1}) = \rho_k V^{(k)}(t_{k_1+1}), \quad 0 < \rho_k < 1. \text{ Then:}$$

$$\begin{aligned} V^{(k)}(t_{k_2+1}) - V^{(k)}(t_{k_1+1}) &= -(1-\rho_k)\, V^{(k)}(t_{k_1+1}) \\ &= -(1-\rho_k)\, \|\boldsymbol{r}(t_{k_1+1})\|^2, \end{aligned}$$

where $1 - \rho_k > 0$. Therefore, the switched system is GAS. ∎

B. Avoiding Zeno Behavior

Given the sequence $T = \{t_0, t_1, \dots, t_n, \dots\}$, $n \in \mathbb{N}$, of switching times, a switched system is Zeno if there exists some finite time t_Z such that:

$$\lim_{n\to\infty} t_n = \sum_{n=0}^{\infty} (t_{n+1} - t_n) = t_Z.$$

In simpler words, Zeno behavior means that the switching times have a finite accumulation point, i.e., that infinite amount of switchings occurs in a finite time interval. In general, the task of detecting possible Zeno trajectories and extending them beyond their accumulation points is far from trivial [8], [14] and depends on the problem at hand.

The proposed OGA policy dictates that a switch among candidate subsystems $\boldsymbol{f}_k$ may occur when $\|\boldsymbol{r}_i(t) - \boldsymbol{r}_j(t)\| \leq R_c$, i.e., when the multi-robot system trajectories $\boldsymbol{r}(t)$ hit the surface $S^c(t) : \{\boldsymbol{r} \in \mathbb{R}^{2\mathrm{N}} \mid \|\boldsymbol{r}_i(t) - \boldsymbol{r}_j(t)\| = R_c\}$. Denote t_d the time instant when the system trajectories lie on the surface $S^c(t_d)$ and the agents are involved in the decision making, t_d^-, t_d^+ the time instants before and after the decision making, respectively, and $k\left(t_d^-\right)$, $k\left(t_d^+\right)$ the assignment before and after the decision making, respectively. The decision making on $S^c(t_d)$ results in two different cases:

1) The agents decide to keep their goal assignment, i.e. $k\left(t_d^-\right) = k\left(t_d^+\right)$, in which case no switching occurs.
2) The agents decide to swap goals, i.e. $k\left(t_d^-\right) \neq k\left(t_d^+\right)$ and a switching occurs.

Theorem 2: The switched multi-robot system (4) under the OGA policy does not suffer from Zeno behavior.

Proof: We employ the results in [15], Theorem 2. Let us assume that the switched system (4) has a Zeno point $\bar{\boldsymbol{r}}$. Then it holds that: $\bar{\boldsymbol{r}} \in S^c(t_d)$ and $\bar{\boldsymbol{r}}$ is an accumulation point of the set $\mathcal{S} = \{\boldsymbol{r} \in S^c(t_d) : \boldsymbol{f}_k\left(t_d^-\right) = \boldsymbol{f}_k\left(t_d^+\right)\}$, where $S^c(t_d)$ is the switching surface at the decision and switching time t_d. The OGA policy dictates that at least one pairwise goal swap occurs at time t_d, since if the agents decided to keep the goals they had at time t_d^-, then t_d would not have been a switching time, a contradiction. Since at least one pair of agents (i, j) switches goal locations, denoted as $\boldsymbol{r}_{G_i}$, $\boldsymbol{r}_{G_j}$, the condition $\boldsymbol{f}_k\left(t_d^-\right) = \boldsymbol{f}_k\left(t_d^+\right)$ on the switched vector fields holds true only when $\boldsymbol{r}_{G_i} = \boldsymbol{r}_{G_j}$ (see also the detailed analysis of Theorem 4 in [11]), which is a contradiction by construction. Thus, the set $\mathcal{S} = \emptyset$, which furthermore implies that the set of its accumulation points is empty, implying that no Zeno points can be contained there. Thus, the switched multi-robot system (4) does not exhibit Zeno behavior. ∎

IV. A Switching Logic on Collision Avoidance

The OGA policy does not ensure that inter-agent collisions, realized as keeping $d_{ij}(t) \geq 2r_0$, $\forall t \in [0, \infty)$, are always avoided. A simple scenario verifying this is given in [11].

Problem 2: We would like to establish (sufficient) conditions under which the OGA policy is collision-free.

A. Detecting Conflicts

Recall that we are referring to time $t > t_d$, i.e. after $\mathrm{N}_c \leq \mathrm{N}$ connected agents have decided on a goal assignment k based on the OGA policy, and move towards their goal locations $\boldsymbol{r}_{gi}^{(k)}$. In the sequel we drop the notation $\cdot^{(k)}$, in the sense that the goal assignment k is kept fixed.

We would first like to identify a metric (a "supervisor") determining online whether the OGA policy results in collisions. Let us consider the collision avoidance constraint:

$$c_{ij}(\boldsymbol{r}_i, \boldsymbol{r}_j) = (x_i - x_j)^2 + (y_i - y_j)^2 - \Delta^2 > 0, \quad (5)$$

encoding that the inter-agent distance $d_{ij} = \|\boldsymbol{r}_i - \boldsymbol{r}_j\|$ should always remain greater than Δ.

To facilitate the analysis using Lyapunov-like approaches, we first need to encode the constraint (5) as a Lyapunov-like function. Inspired by interior point methods we first define the logarithmic barrier function $b_{ij}(\cdot) : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ for the constraint (5) as:

$$b_{ij}(\boldsymbol{r}_i, \boldsymbol{r}_j) = -\ln\left(c_{ij}(\boldsymbol{r}_i, \boldsymbol{r}_j)\right), \quad (6)$$

which tends to $+\infty$ as $c_{ij}(\cdot) \to 0$, i.e. as $d_{ij} \to \Delta$. The recentered barrier function of (6) is defined as [16]:

$$r_{ij} = b_{ij}(\boldsymbol{r}_i, \boldsymbol{r}_j) - b_{ij}(\boldsymbol{r}_g, \boldsymbol{r}_j) - \nabla b_{ij}^T\Big|_{\boldsymbol{r}_g}(\boldsymbol{r}_i - \boldsymbol{r}_g), \quad (7)$$

where $\boldsymbol{r}_g = \begin{bmatrix} x_g & y_g \end{bmatrix}^T$ is a desired set-point (i.e., goal location in our problem) within the collision-free space,

$b_{ij}(\boldsymbol{r}_g, \boldsymbol{r}_j)$ is the value of (6) evaluated at $\boldsymbol{r}_g$, $\nabla b_{ij} = \begin{bmatrix} \frac{\partial b_{ij}}{\partial x_i} & \frac{\partial b_{ij}}{\partial y_i} \end{bmatrix}^T$ is the gradient vector of the function $b_{ij}(\cdot)$, and $\nabla b_{ij}{}^T\big|_{\boldsymbol{r}_g}$ is the transpose of the gradient vector evaluated at the goal position $\boldsymbol{r}_g$. By construction, the recentered barrier function (7):

(i) is non-zero everywhere except for the goal location $\boldsymbol{r}_g$,
(ii) tends to $+\infty$ as the distance d_{ij} tends to Δ.

These characteristics of recentered barrier functions (7) are suitable for encoding *both* collision avoidance of each agent i w.r.t. an agent $j \neq i$, *and* convergence of agent i to its assigned goal $\boldsymbol{r}_{g_i}$. To ensure that we have a nonnegative function encoding these objectives, so that it can be used as a Lyapunov-like function, for each agent i we define [10]:

$$w_{ij}(\cdot) = \left(r_{ij}(\boldsymbol{r}_i, \boldsymbol{r}_j, \boldsymbol{r}_{g_i})\right)^2, \tag{8}$$

which now is a positive definite function. To furthermore encode that the position trajectories of agent i remain bounded in a prescribed region (for reasons that will be explained in the technical analysis later on), for each agent i we define the "workspace" constraint:

$$c_{i0}(\boldsymbol{r}_i, \boldsymbol{r}_0) = R_0^2 - (x_i - x_0)^2 - (y_i - y_0)^2 > 0, \tag{9}$$

which encodes that the position $\boldsymbol{r}_i$ should always lie in the interior of a circle of center $\boldsymbol{r}_0 = \begin{bmatrix} x_0 & y_0 \end{bmatrix}^T$ and radius $R_0 > R_c$. This can be thought as the workspace where the agents operate. Then, in the same reasoning followed before, we define the barrier function:

$$b_{i0}(\boldsymbol{r}_i, \boldsymbol{r}_0) = -\ln\left(c_{i0}(\boldsymbol{r}_i, \boldsymbol{r}_0)\right), \tag{10}$$

and its corresponding recentered barrier function:

$$r_{i0} = b_{i0}(\boldsymbol{r}_i, \boldsymbol{r}_0) - b_{i0}(\boldsymbol{r}_{g_i}, \boldsymbol{r}_0) - \nabla b_{i0}{}^T\big|_{\boldsymbol{r}_{g_i}}(\boldsymbol{r}_i - \boldsymbol{r}_{g_i}), \tag{11}$$

which vanishes only at the goal location $\boldsymbol{r}_{g_i}$ and tends to infinity as the position $\boldsymbol{r}_i$ tends to the workspace boundary. To get a positive definite function we consider:

$$w_{i0}(\cdot) = \left(r_{i0}(\boldsymbol{r}_i, \boldsymbol{r}_0, \boldsymbol{r}_{g_i})\right)^2. \tag{12}$$

Therefore, an encoding that agent i stays Δ apart w.r.t. all of its neighbor agents, while staying within the bounded workspace, can be now given by an approximation of the maximum function of the form [10]:

$$w_i = \left((w_{i0})^\delta + \sum_{j \in \mathcal{N}_c} (w_{ij})^\delta\right)^{\frac{1}{\delta}}, \tag{13}$$

where $\delta \in [1, \infty)$, and $\mathcal{N}_c \subseteq \mathcal{N}$ is the set of neighbor agents $j \neq i$ of agent i. Finally, to ensure that we have a Lyapunov-like function for agent i which uniformly attains its maximum value on constraints' boundaries we take:

$$W_i = \frac{w_i}{1 + w_i}, \tag{14}$$

which is zero for $w_i = 0$, i.e., at the goal position $\boldsymbol{r}_{g_i}$ of agent i, and equal to 1 as $w_i \to \infty$. For more details on the analytical construction the reader is referred to [10].

The Lyapunov-like function (14) can now be used as a (sufficient) criterion on determining whether the control inputs $\boldsymbol{u}_i$, $\boldsymbol{u}_j$ of the OGA policy jeopardize safety. Let us consider the time derivative of (14):

$$\begin{aligned} \dot{W}_i &= \begin{bmatrix} \frac{\partial W_i}{\partial x_i} & \frac{\partial W_i}{\partial y_i} \end{bmatrix} \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} + \sum_{j \in \mathcal{N}_c} \left(\begin{bmatrix} \frac{\partial W_i}{\partial x_j} & \frac{\partial W_i}{\partial y_j} \end{bmatrix} \begin{bmatrix} \dot{x}_j \\ \dot{y}_j \end{bmatrix} \right) = \\ &= \boldsymbol{\zeta}_i{}^T \boldsymbol{u}_i + \sum_{j \in \mathcal{N}_c} \left(\boldsymbol{\zeta}_{ij}{}^T \boldsymbol{u}_j \right), \end{aligned} \tag{15}$$

where $\boldsymbol{\zeta}_i \triangleq \begin{bmatrix} \frac{\partial W_i}{\partial x_i} & \frac{\partial W_i}{\partial y_i} \end{bmatrix}^T$, $\boldsymbol{\zeta}_{ij} \triangleq \begin{bmatrix} \frac{\partial W_i}{\partial x_j} & \frac{\partial W_i}{\partial y_j} \end{bmatrix}^T$. The evolution of the time derivative $\dot{W}_i$ along the trajectories of agent i depends not only on its own motion (through $\boldsymbol{u}_i$), but also on the motion of its neighbor agents $j \neq i$ through their velocities $\boldsymbol{u}_j$. *This time derivative provides sufficient conditions on establishing that the* OGA *policy is safe*. To see how, let us consider the following Lemma.

Lemma 1: If the control inputs $\boldsymbol{u}_i$, $\boldsymbol{u}_j$, where $j \in \mathcal{N}_c$, out of the OGA policy satisfy the following condition for all $i \in \{1, \ldots, \mathrm{N}\}$:

$$\boldsymbol{\zeta}_i{}^T \boldsymbol{u}_i + \sum_{j \in \mathcal{N}_c} \left(\boldsymbol{\zeta}_{ij}{}^T \boldsymbol{u}_j \right) < 0, \tag{16}$$

then the OGA policy is asymptotically stable w.r.t. the current goal assignment k and furthermore collision-free.

Proof: To verify the argument, consider the properties of the Lyapunov-like function (14) and denote the constrained set for each agent i as $\mathcal{K}_i = \{\boldsymbol{r} \in \mathbb{R}^{2\mathrm{N}} \mid c_{ij}(\cdot) \geq 0\}$, where $j \in \mathcal{N}_c \cup \{0\}$. The set $\mathcal{K}_i$ is by construction compact (i.e., closed and bounded), with the level sets of W_i being closed curves contained in the set $\mathcal{K}_i$. Then, the condition (16) implies that the system trajectories $\boldsymbol{r}_i(t)$ under the OGA control input $\boldsymbol{u}_i$ evolve downwards the level sets of W_i, i.e. always remain in the interior of the constrained set $\mathcal{K}_i$, which furthermore reads that the inter-agent distances $d_{ij}(t)$ never violate their critical value Δ. ■

Problem 3: We need to establish a LR policy ensuring that inter-agent distances d_{ij} remain greater than a critical distance Δ, and will be active only when the OGA policy is about to result in colliding trajectories.

B. Resolving Conflicts

The condition (16) gives rise to furthermore determining a LR control policy in the case that the OGA policy is about to violate it. Let us denote the switching surface:

$$\mathcal{Q}_i = \boldsymbol{\zeta}_i{}^T \boldsymbol{u}_i + \sum_{j \in \mathcal{N}_c} \left(\boldsymbol{\zeta}_{ij}{}^T \boldsymbol{u}_j \right). \tag{17}$$

Then for $\mathcal{Q}_i < 0$ one has out of Lemma 1 that the OGA policy is both GAS and collision-free. For $\mathcal{Q}_i > 0$ one has $\dot{W}_i > 0$, which implies that the position trajectories $\boldsymbol{r}_i(t)$ evolve upwards the level sets of the Lyapunov-like function W_i. This condition *may* jeopardize safety and dictates the definition of an additional, LR policy, which will ensure that the trajectories $\boldsymbol{r}_i(t)$ remain in the constrained set $\mathcal{K}_i$.

Lemma 2: The LR control policy for agent i, realized via:

$$\boldsymbol{u}_{ib} = \begin{bmatrix} \frac{-k_i W_i - \sum_{j \in \mathcal{N}_c} \frac{\partial W_i}{\partial x_j} u_{jx}}{\frac{\partial W_i}{\partial x_i}} & \frac{-k_i W_i - \sum_{j \in \mathcal{N}_c} \frac{\partial W_i}{\partial y_j} u_{jy}}{\frac{\partial W_i}{\partial y_i}} \end{bmatrix}^T, \tag{18}$$

where $k_i > 0$, renders the trajectories of the multi-robot system collision-free in the set $\mathcal{Q}_i > 0$.

Proof: To verify the argument, consider that the time derivative of the Lyapunov-like function (15) under the proposed control inputs (18) reads: $\dot{W}_i = -k_i W_i$, which reads that the position trajectories $\boldsymbol{r}_i(t)$ move downwards the level sets of the Lyapunov-like function (14). ■

C. Treating an ill-conditioned case

The recentered barrier function (7) is undefined in the case that an agent j lies in the vicinity of the goal location of agent i. This situation may occur throughout the system evolution and renders both the condition (16) and the control law (18) undefined. Thus, an additional control policy is needed to take action in such cases. To this end, first we modify (7) by canceling out the $b_{ij}(\boldsymbol{r}_{g_i}, \boldsymbol{r}_g)$ term, to get:

$$r_{ij}{}^m = b_{ij}(\boldsymbol{r}_i, \boldsymbol{r}_j) - \nabla b_{ij}{}^T\big|_{\boldsymbol{r}_g}(\boldsymbol{r}_i - \boldsymbol{r}_g), \tag{19}$$

and following the same steps described above we construct the function $W_i{}^m$. *This function is not positive definite w.r.t. the goal location* $\boldsymbol{r}_{g_i}$ and thus can not be used as a (generalized) Lyapunov-like function for establishing convergence to the goal location, yet it encodes that agent i should stay Δ apart from any neighbor agent $j \neq i$ and also stay within the workspace boundary. This gives rise to the control law:

$$\boldsymbol{u}_{ib} = \begin{bmatrix} \frac{-k_i W_i{}^m - \sum_{j\in\mathcal{N}_c} \frac{\partial W_i{}^m}{\partial x_j} u_{jx}}{\frac{\partial W_i{}^m}{\partial x_i}} & \frac{-k_i W_i^m - \sum_{j\in\mathcal{N}_c} \frac{\partial W_i{}^m}{\partial y_j} u_{jy}}{\frac{\partial W_i{}^m}{\partial y_i}} \end{bmatrix}^T \tag{20}$$

which yields: $\dot{W}_i{}^m = -k_i W_i{}^m$, implying thus that the trajectories $\boldsymbol{r}_i(t)$ evolve within the constrained set $\mathcal{K}_i$ of agent i. This control law is activated only when (16) becomes ill-conditioned. We call this policy the modified LR policy.

D. Ensuring both Stability and Safety

The switching between the OGA policy and LR policy across the switching surface $\mathcal{Q}_i = 0$, as well as the switching between the LR policy and modified LR policy when equation (16) becomes ill-conditioned give rise to dynamics with discontinuous right-hand side for each agent i:

$$\dot{\boldsymbol{r}}_p(t) = \boldsymbol{v}_i^{(p)}, \tag{21}$$

Consider the switching sequence of times $\mathcal{T} = \{\tau_1, \tau_2, \tau_3, \ldots, \tau_n, \ldots,\}$, where the time interval $\tau_n - \tau_{n-1}$ is finite $\forall n \in \mathbb{N}$, excluding thus any sliding-like behavior (chattering), and the switching sequence:

$$\Sigma^\star = \{\boldsymbol{r}_i(\tau_0); (p_0, \tau_0), (p_1, \tau_1), \ldots, (p_n, \tau_n), \ldots,\},$$

where $\tau_0 > t_d$, $q \in \mathbb{N}$ and $p_n \in \mathcal{P}$. Note that the switching between the three candidate policies *is not necessarily periodic*.

Problem 4: We finally need to establish that the proposed switching strategy $\Sigma^\star$ between the OGA policy, the LR policy and the modified LR policy renders the multi-robot trajectories stable w.r.t. to the assigned goals and also collision-free. This switching strategy makes it rather difficult to check Branicky's decreasing condition on the "switched-on" time intervals of each individual subsystem. For this reason we resort to results that remove this condition in order to draw conclusions on (asymptotic) stability.

Theorem 3: The trajectories of the switched multi-robot system under the switching logic $\Sigma^\star$ are (i) collision-free, and (ii) asymptotically stable w.r.t. the assigned goals.

Proof: The first argument is proved in Lemmas 1, 2 and in Section IV-C. The second argument can be verified by applying [17, Theorem 3.9]. Lemmas 1, 2 imply that the functions V_i, W_i for each agent i serve as generalized Lyapunov-like functions for the individual subsystems $p \in \{1, 2\}$, respectively. The stability condition [17, (5) in Theorem 3.9] for each subsystem p is satisfied out of the boundedness of the solutions $\boldsymbol{r}_i(t)$ within the constrained set $\mathcal{K}_i$, which is by construction closed and bounded. Therefore, the switched system is stable. To furthermore establish asymptotic stability, it should hold that for at least one of the individual subsystems p, the value of the corresponding generalized Lyapunov-like function decreases along the sequence of switching times. Let us assume that this is not the case; then this would imply that the closed-loop switched vector fields $\boldsymbol{v}_i^{(p)}$, $p \in \{1, 2\}$, "oppose" each other and cancel out on the switching surface $\mathcal{Q}_i$, forcing the system trajectories to get stuck there. Note that the vector field $\boldsymbol{v}_i^{(2)} = \boldsymbol{u}_{ib}$ is by construction parallel to the gradient vector of the function W_i, which is by construction transverse to the boundary of the constrained set $\mathcal{K}_i$.[2] Then, this assumption would furthermore imply that the vector field $\boldsymbol{v}_i^{(1)} = \boldsymbol{u}_i$ of the OGA policy forces the system trajectories of agent i towards a goal on an intersecting path w.r.t. the neighbor agent; a contradiction, since the OGA does not result in intersecting paths [1]. For the subsystem $p = 3$ we consider the generalized Lyapunov function $X_i = \|\boldsymbol{r}_i - \boldsymbol{r}_{g_i}\|^2$. The modified LR policy dictates that agent i is repelled from agent j, only when agent j lies on the goal position $\boldsymbol{r}_{g_i}$. This implies that the value of the Lyapunov-like function X_i on the corresponding time interval increases. By construction, one has that this growth is bounded from above, since the trajectories $\boldsymbol{r}_i(t)$ are confined within the compact constrained set K_i. To ensure asymptotic stability for the switched multi-robot system, we furthermore need to ensure that the system will switch to either $p = 1$ or $p = 2$. This is ensured since agent j eventually leaves the goal location $\boldsymbol{r}_{g_i}$, moving towards its own goal location $\boldsymbol{r}_{g_j}$, which implies that the modified LR policy is no longer needed and the system switches either to $p = 1$ or $p = 2$, depending on the condition (16). Under this caveat, i.e. that agent j is stuck on $\boldsymbol{r}_{g_i}$ for some reason is highly unlikely, one has that the switched multi-robot system under $\Sigma^\star$ is asymptotically stable. ■

V. Simulation Results

Simulation results are provided to evaluate the performance of the switched multi-robot system under the proposed switching logic and control policies. Let us here consider a

[2] This is ensured since the function W_i takes uniformly its maximum value on the boundary of the constrained set $\mathcal{K}_i$.

typical scenario involving N = 10 agents with initial and goal locations in very close proximity such that the OGA policy is not sufficient to ensure there are no collisions, see Fig. 1. The agents start from the initial conditions marked with red squares towards goal locations that do not necessarily correspond to the optimal assignment. The paths from the starting locations to the initially assigned goals are depicted as the blue lines. Communications range is denoted by the cyan ring around each agent and in this case $R_c = 5R$ During this simulation, at least one subgroup of robots using using the LR control policy for 21% of the duration of the simulation. Fig. 2 shows the minimum clearance between any two robots. As this is always positive, there is never a collision between any robots.

The number of re-plan operations depends greatly on the number of robots, the communications range, initial distribution, and quality of initial assignments. The time for a re-plan operation also directly depends on the number of agents in the connected component. Planning times on simulated teams of robots using a single computer take approximately $10^{-7} {N_c}^3$ seconds such that a component of 100 robots takes about 0.1 seconds to plan. This is more than sufficient for the proposed applications.

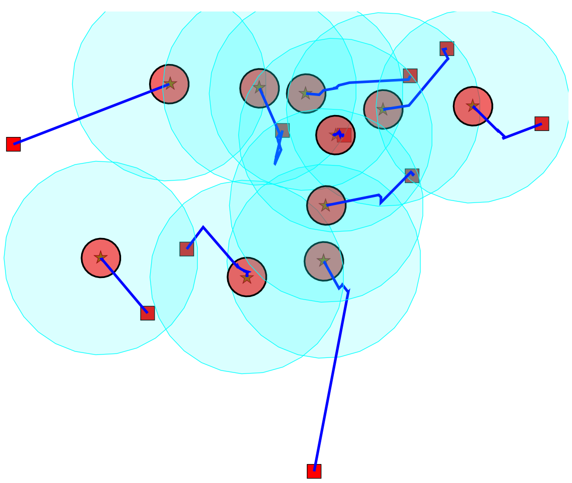

Fig. 1. A simulation with $N = 10$ robots. The paths followed are straight when in the OGA segments, but are potentially curved when using the LR control law.

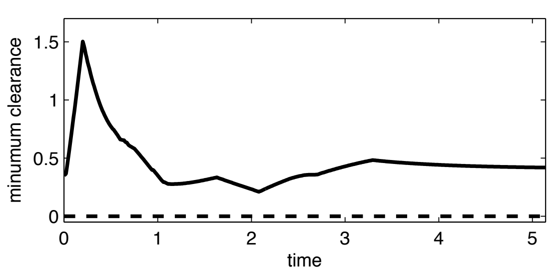

Fig. 2. The minimum clearance between any two robots for a simulated trial with $N = 10$, where clearance is defined as the free space between robots. Note that as this is always positive, there were never any collisions between robots.

VI. Discussion and Concluding Remarks

We presented a switched systems approach on the decentralized concurrent goal assignment and trajectory generation for multi-robot networks, which guarantees safety and global stability to interchangeable goal locations. The proposed switching logic relies on multiple Lyapunov-like functions which encode goal swap among locally connected agents, avoidance of inter-agent collisions and convergence to the assigned goal locations. As such, the proposed methodology renders feedback control policies with local coordination only, and is therefore suitable for applications such as first-response deployment of robotic networks under limited information sharing. Our current work focuses on the consideration of agents with more complicated dynamics, as well as the robustness of our algorithms under communication failures and uncertainty.

References

[1] M. Turpin, N. Michael, and V. Kumar, "Trajectory planning and assignment in multirobot systems," in *Proc. of Int. Workshop on the Algorithmic Foundations of Robotics (WAFR)*, Boston, MA, Jun. 2012.

[2] S. Kloder and S. Hutchinson, "Path planning for permutation-invariant multirobot formations," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 650–665, Aug. 2006.

[3] G. Arslan, J. R. Marden, and J. S. Shamma, "Autonomous vehicle-target assignment: A game-theoretical formulation," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 129, pp. 584–596, Sep. 2007.

[4] M. M. Zavlanos and G. J. Pappas, "Dynamic assignment in distributed motion planning with local coordination," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 232–242, Feb. 2008.

[5] S. L. Smith and F. Bullo, "Monotonic target assignment for robotic networks," *IEEE Trans. on Automatic Control*, vol. 54, no. 9, pp. 2042–2057, Sep. 2009.

[6] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912–926, Aug. 2009.

[7] L. Liu and D. Shell, "Multi-level partitioning and distribution of the assignment problem for large-scale multi-robot task allocation," in *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2011.

[8] D. Liberzon, *Switching in Systems and Control*. Birkhauser, 2003.

[9] M. S. Branicky, "Multiple Lyapunov functions and other analysis tools for switched and hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 475–482, Apr. 1998.

[10] D. Panagou, D. M. Stipanović, and P. G. Voulgaris, "Multi-objective control for multi-agent systems using Lyapunov-like barrier functions," in *Proc. of the 52nd IEEE Conference on Decision and Control*, Florence, Italy, Dec. 2013, pp. 1478–1483.

[11] D. Panagou, M. Turpin, and V. Kumar, "Decentralized goal assignment and trajectory generation in multi-robot networks (technical report)." [Online]. Available: http://arxiv.org/

[12] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial & Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[13] S. Giordani, M. Lujak, and F. Martinelli, "A distributed algorithm for the multi-robot task allocation problem," in *Trends in Applied Intelligent Systems*. Springer, 2010, pp. 721–730.

[14] A. D. Ames and S. Sastry, "Characterization of Zeno behavior in hybrid systems using homological methods," in *Proc. of the 2005 American Control Conference*, Portland, OR, USA, Jun. 2005, pp. 1160–1165.

[15] F. Ceragioli, "Finite valued feedback laws and piecewise classical solutions," *Nonlinear Analysis*, vol. 65, no. 5, pp. 984–998, 2006.

[16] A. G. Wills and W. P. Heath, "A recentred barrier for constrained receding horizon control," in *Proc. of the American Control Conference*, Anchorage, AK, USA, May 2002, pp. 4177–4182.

[17] J. Zhao and D. J. Hill, "On stability, L_2-gain and H_∞ control for switched systems," *Automatica*, vol. 44, pp. 1220–1232, 2008.