

# Time-constrained multi-agent task scheduling based on prescribed performance control

Pian Yu and Dimos V. Dimarogonas

**Abstract**—The problem of time-constrained multi-agent task scheduling and control synthesis is addressed. We assume the existence of a high level plan which consists of a sequence of cooperative tasks, each of which is associated with a deadline and several Quality-of-Service levels. By taking into account the reward and cost of satisfying each task, a novel scheduling problem is formulated and a path synthesis algorithm is proposed. Based on the obtained plan, a distributed hybrid control law is further designed for each agent. Under the condition that only a subset of the agents are aware of the high level plan, it is shown that the proposed controller guarantees the satisfaction of time constraints for each task. A simulation example is given to verify the theoretical results.

## I. INTRODUCTION

The integration of multi-agent cooperative control and task scheduling is of great practical interest for applications such as robotics. Over the past decades, the research in multi-agent cooperative control has usually focused on achieving one single global task, such as reference-tracking [1], consensus [2] or formation [3]. In practice, a group of agents encounters the request of a sequence of tasks. Furthermore, deadline constraints on the completion of each task is a common requirement, e.g., “Visit region A within 10 time units”. How to jointly schedule the time-constrained task sequence and design the distributed controllers for the group of agents is a more recent challenge.

Task scheduling is one of the fundamental issues in the area of real-time systems [4]. The scheduling algorithms can be divided into two categories: static and dynamic scheduling. Rate monotonic (RM) scheduling [5] and its extensions [6] are static scheduling algorithms and represent one major paradigm, while many dynamic scheduling algorithms are based on Earliest Deadline First (EDF) policies [5], which represent a second major paradigm. Under certain conditions, EDF has been shown to be optimal in resource-efficient environments [7]. However, these scheduling algorithms usually do not consider the reward or/and cost of completion each task. Different from the above algorithms, in this paper, a scheduling algorithm is proposed for a sequence of tasks by taking into account the reward and cost of completing each task. Motivated by [16], the reward is defined based on the Quality-of-Service (QoS) level, which is determined by the completion time. Moreover, the cost is defined as the (estimated) total distance travelled by the group of agents.

This work was supported in part by the Swedish Research Council (VR), the Swedish Foundation for Strategic Research (SSF), the Knut and Alice Wallenberg Foundation (KAW) and the SRA TNG ICT project TOUCHES.

The authors are with School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, 10044 Stockholm, Sweden. [pianyu@kth.se](mailto:pianyu@kth.se), [dimos@kth.se](mailto:dimos@kth.se)

Cooperative control of multi-agent systems (MAS) has traditionally focused on designing local control laws to achieve a global control objective. Recently, prescribed performance control (PPC) [8] has been proposed to tackle multi-agent control problems with transient performance constraints. In [9], [10], consensus of MAS with prescribed performance on the position error or combined error was investigated. In [11], [12], formation control of MAS and large vehicular platoons were investigated with prescribed transient and steady-state performance. In [14], PPC was utilized to satisfy temporal logic tasks for MAS. In our work, to guarantee that each task is completed at certain QoS level, it is required that each task is completed at a specific time interval, for example, “Visit region A within 6 to 8 time units”. The intuition for the use of PPC is that the time interval constraints under consideration can actually be translated into transient performance constraints, and thus PPC can be applied.

Motivated by the above discussion, this paper investigates the problem of time-constrained multi-agent task scheduling and control synthesis. The contributions of the paper can be summarized as: i) a novel scheduling problem is formulated for MAS subject to a sequence of cooperation tasks, where each task is associated with a deadline and several QoS levels; ii) under the condition that only a subset of agents are aware of the high level plan, a distributed hybrid control law is designed for each agent that guarantees the achievement of each task, according to certain time interval constraints.

The rest of the paper is organized as follows. In Section II, notation and preliminaries are introduced, while Section III formalizes the considered problem. Section IV presents the proposed solution in detail, which is further verified by simulations in Section V. Conclusions are given in Section VI.

## II. PRELIMINARIES

### A. Notation

Let  $\mathbb{R} := (-\infty, \infty)$ ,  $\mathbb{R}_{\geq 0} := [0, \infty)$ ,  $\mathbb{R}_{> 0} := (0, \infty)$ ,  $\mathbb{Z}_{> 0} := \{1, 2, \dots\}$  and  $\mathbb{Z}_{\geq 0} := \{0, 1, 2, \dots\}$ . Denote  $\mathbb{R}^n$  as the  $n$  dimensional real vector space,  $\mathbb{R}^{n \times m}$  as the  $n \times m$  real matrix space.  $I_n$  is the identity matrix of order  $n$ . For  $(x_1, x_2, \dots, x_m) \in \mathbb{R}^{n_1+n_2+\dots+n_m}$ , the notation  $(x_1, x_2, \dots, x_m)$  stands for  $[x_1^T, x_2^T, \dots, x_m^T]^T$ . Let  $|\lambda|$  be the absolute value of a real number  $\lambda$ ,  $\|x\|$  and  $\|A\|$  be the Euclidean norm of vector  $x$  and matrix  $A$ , respectively. For a set  $\Omega$ ,  $|\Omega|$  represents the cardinality of  $\Omega$ . In addition, we use  $\cap$  to denote the set intersection and  $\cup$  the set union.  $P \succ 0$  means that  $P$  is a positive definite matrix and  $P^T$  is the transpose of  $P$ . The Kronecker product is denoted by  $\otimes$ .

## B. Graph Theory

Let  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$  be a graph with the set of nodes  $\mathcal{V} = 1, 2, \dots, N$ , and  $\mathcal{E} \subseteq \{(i, j) : i, j \in \mathcal{V}, j \neq i\}$  the set of edges. If  $(i, j) \in \mathcal{E}$ , then node  $j$  is called a neighbor of node  $i$  and node  $j$  can receive information from node  $i$ . The neighboring set of node  $i$  is denoted by  $\mathcal{N}_i = \{j \in \mathcal{V} | (j, i) \in \mathcal{E}\}$ .

A graph is called undirected if  $(i, j) \in \mathcal{E} \Leftrightarrow (j, i) \in \mathcal{E}$ , and a graph is connected if for every pair of nodes  $(i, j)$ , there exists a path which connects  $i$  and  $j$ , where a path is an ordered list of edges such that the head of each edge is equal to the tail of the following edge.

## III. PROBLEM FORMULATION

### A. Agent dynamics

Consider a group of  $N$  agents, each of which obeys the second-order dynamics:

$$\begin{aligned} \dot{x}_i(t) &= v_i(t), \\ \dot{v}_i(t) &= u_i(t), \quad i = 1, 2, \dots, N. \end{aligned} \quad (1)$$

where  $x_i \in \mathbb{R}^n, v_i \in \mathbb{R}^n$  and  $u_i \in \mathbb{R}^n$  are the position, velocity and control input of agent  $i$ , respectively. Let  $x = (x_1, \dots, x_N), v = (v_1, \dots, v_N)$  be the stack vector of positions and velocities, respectively. Denote by  $\bar{x} = (\bar{x}_1, \dots, \bar{x}_m)$  the  $m$ -dimensional stack vector of relative positions of pairs of agents that form an edge in  $\mathcal{G}$ , where  $m$  is the number of edges. The elements of vector  $\bar{x}$  are defined by  $\bar{x}_k \triangleq x_{ij} = x_i - x_j$ , with  $k \in \{1, 2, \dots, m\}$ . It is assumed that the graph  $\mathcal{G}$  is undirected and connected.

### B. Task Specifications

For the group of agents, we assume the existence of a high level plan  $\phi$  given and known by only a subset of the agents that consists of the achievement of a sequence of cooperative tasks. Then, we can denote  $\phi$  as a sequence of  $M \in \mathbb{Z}_{>0}$  objectives  $\phi_1, \phi_2, \dots, \phi_M$ . Note that the order at which the tasks should be completed is not predefined in this paper.

To be more specific, we consider that the group of agents have a set of regions of interest, denoted by  $\mathbb{X} \triangleq \{\mathbb{X}_1, \dots, \mathbb{X}_M\}$ , where  $\mathbb{X}_l, l = 1, \dots, M$  is the target region for task  $\phi_l$ . For simplicity of presentation,  $\mathbb{X}_l \in \mathbb{X}$  is represented by a sphere area around a point of interest:

$$\mathbb{X}_l = \mathcal{B}(c_l, r_l) = \{z \in \mathbb{R}^n : \|z - c_l\| \leq r_l\},$$

where  $c_l \in \mathbb{R}^n$  is the center,  $r_{\min} \leq r_l \leq r_{\max} \in \mathbb{R}$  is the radius and  $0 < r_{\min} < r_{\max}$  are the upper and lower bounds of the radius for all regions. Define  $\mathbb{X}_0 := \{x_1(t_0), \dots, x_N(t_0)\}$ , which represents a collection of the agents' initial states.

*Assumption 1:* The initial state set and target sets do not intersect, i.e.,  $\mathbb{X}_{l_1} \cap \mathbb{X}_{l_2} = \emptyset, \forall l_1, l_2 \in \{0, 1, \dots, M\}, l_1 \neq l_2$ .

Each task has several Quality-of-Service (QoS) levels, which are determined by the (actual) completion time. In this task model, each task  $\phi_l$  is characterized by the following parameters:

- $D_l$ : the absolute deadline;
- $k_l$ : number of QoS levels ( $k_l \in \mathbb{Z}, k_l \geq 2$ );
- $AE_l$ : the (actual) completion time;

In addition, for each QoS level  $\hat{k}_l, \hat{k}_l \in \{0, 1, \dots, k_l - 1\}$ , there are two more parameters:

$EE_l[\hat{k}_l]$ : the estimated completion time of task  $\phi_l$  at QoS level  $\hat{k}_l$ ;

$R_l[\hat{k}_l]$ : the reward that task  $\phi_l$  contributes if it is completed at QoS level  $\hat{k}_l$ . QoS level 0 represents the rejection of the task and  $R_l[0] < 0$  is called the rejection penalty [15], [16].

The QoS levels are defined as follows: for each task  $\phi_l$ , divide the time interval  $[t_0, D_l]$  into  $k_l - 1$  disjoint parts  $(t_l^1, D_l], (t_l^2, t_l^1], \dots, (t_0, t_l^{k_l-2}]$  (not necessarily equal division), such that  $D_l := t_l^0 > t_l^1 > t_l^2 > \dots > t_l^{k_l-1} := t_0$  (see Fig.1). We say that the task  $\phi_l$  is completed at QoS level 0, if  $AE_l > D_l$ , and we say that the task  $\phi_l$  is completed at QoS level  $\hat{k}_l, \hat{k}_l \geq 1$ , if  $AE_l \in (t_l^{\hat{k}_l}, t_l^{\hat{k}_l-1}]$ . In this paper, it is assumed that the QoS levels and the corresponding time intervals are known to each agent. In addition, without loss of generality, we assume  $t_0 = 0, \forall l$ .

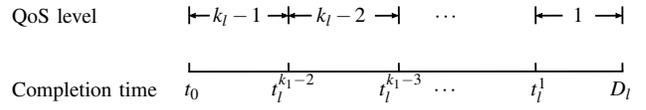


Fig. 1. Completion time and the corresponding QoS level

### C. Reward and cost of a path

In this paper, we are interested in the quantitative reward and cost of satisfying the task plan  $\phi$ . Let  $\Pi_\phi$  be the set of permutations of the set  $\{1, \dots, M\}$  and  $\pi \in \Pi_\phi$  be  $\pi := \{\pi[1], \pi[2], \dots, \pi[M]\}$ . Denote by  $\mathbf{P}(\pi) = \mathbf{P}(\pi[0]\pi[1]\pi[2] \dots \pi[M]) = \mathbb{X}_{\pi[0]}\mathbb{X}_{\pi[1]}\mathbb{X}_{\pi[2]} \dots \mathbb{X}_{\pi[M]}$ , a path generated by the task set  $\phi$  with the order of completion given by  $\pi$ , where  $\pi[0] = 0, \forall \pi \in \Pi_\phi$ .

1) *Reward:* The reward of a path  $\mathbf{P}(\pi)$  is given by

$$R(\mathbf{P}(\pi)) = \sum_{l=1}^M R_{\pi[l]}[\hat{k}_{\pi[l]}]. \quad (2)$$

2) *Cost:* Motivated by [17], the cost of the path  $\mathbf{P}(\pi)$  is defined as the (estimated) distance travelled by the group of agents. Let  $W(\mathbb{X}_{\pi[k]}, \mathbb{X}_{\pi[k+1]}) : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$  be the transition cost from set  $\mathbb{X}_{\pi[k]}$  to  $\mathbb{X}_{\pi[k+1]}$ , which is given by

$$W(\mathbb{X}_{\pi[k]}, \mathbb{X}_{\pi[k+1]}) \triangleq N(\|c_{\pi[k]} - c_{\pi[k+1]}\|), \quad k = 1, \dots, M - 1,$$

$$W(\mathbb{X}_{\pi[0]}, \mathbb{X}_{\pi[1]}) \triangleq \sum_{i=1}^N \|x_i(0) - c_{\pi[1]}\|.$$

Then, the cost of  $\mathbf{P}(\pi)$  is defined as:

$$C(\mathbf{P}(\pi)) = \sum_{k=0}^{M-1} W(\mathbb{X}_{\pi[k]}, \mathbb{X}_{\pi[k+1]}). \quad (3)$$

It is obvious that the completion of the sequence of tasks can be scheduled in different orders, which correspond to different paths. Let  $\mathbb{P} = \cup_{\pi \in \Pi} \{\mathbf{P}(\pi)\}$  be the set of all paths. In this paper, we want to maximize the reward as well as minimize the cost. Therefore, we propose the following objective function

$$\mathbf{J}(\mathbb{P}) \triangleq \max_{\pi \in \Pi} \{\alpha R(\mathbf{P}(\pi)) - (1 - \alpha)C(\mathbf{P}(\pi))\}, \quad (4)$$

where  $\alpha \in [0, 1]$  is a parameter used to balance between the reward and the cost.

Since the high level plan  $\phi$  is known to only a subset of the agents, it is necessary for the group of agents to coordinate to complete the tasks. Here, we divide the group of agents into two different groups, active agents (agents who know where the target regions are) and passive agents (agents who do not know where the target regions are). The purpose of this paper is to maximizing the objective function (4) for the set of tasks  $\phi$ . Formally, the problem is stated below.

*Problem 1:* Given a group of  $N$  agents (which are divided into active agents and passive agents), whose dynamics is given in (1), and the task specifications in Section III-B, design distributed (i.e., with only measurements of states of neighbors) control laws  $u_i$  and the associated path  $\mathbf{P}(\pi)$ , such that (4) is maximized by  $\mathbf{P}(\pi)$ .

#### IV. SOLUTION

The proposed solution consists of two layers: i) an offline path synthesis layer, i.e., progressive goal regions and ii) a distributed control law that guarantees that the group of agents (both active and passive) arrive at their progressive goal regions before the corresponding deadline at all times.

##### A. Path synthesis

As stated previously, the completion of the sequence of tasks can be scheduled in different order. To find the one that maximizes (4), we propose the following scheduling algorithm  $\mathcal{P}(\phi)$ :

$$\max_{\pi \in \Pi_\phi, \hat{k}_\pi} \alpha \sum_{i=1}^{|\pi|} R_{\pi[i]} - (1 - \alpha) \sum_{k=1}^{|\pi|-1} W(\mathbb{X}_{\pi[k]}, \mathbb{X}_{\pi[k+1]}) \quad (5a)$$

subject to

$$\hat{k}_{\pi[i]} \in \{0, 1, \dots, k_{\pi[i]} - 1\}, \quad (5b)$$

$$R_{\pi[i]} = R_{\pi[i]}[\hat{k}_{\pi[i]}], \quad (5c)$$

$$EE_{\pi[i]}[0] = D_{\pi[i]} + \varepsilon, \quad (5d)$$

$$EE_{\pi[i]}[\hat{k}_{\pi[i]}] = t_{\pi[i]}^{\hat{k}_{\pi[i]}-1}, \hat{k}_{\pi[i]} \geq 1 \quad (5e)$$

$$EE_{\pi[i+1]}[\hat{k}_{\pi[i+1]}] > EE_{\pi[i]}[\hat{k}_{\pi[i]}], i = 1, \dots, |\pi| - 1. \quad (5f)$$

where

$$\varepsilon = \min \left\{ \left| t_{l_1}^{\hat{k}_{l_1}} - t_{l_2}^{\hat{k}_{l_2}} \right| : l_1, l_2 \in \{1, \dots, M\}, \hat{k}_{l_1} \in \{0, \dots, k_{l_1} - 2\}, \right. \\ \left. \hat{k}_{l_2} \in \{0, \dots, k_{l_2} - 2\}, l_1 \neq l_2, t_{l_1}^{\hat{k}_{l_1}} \neq t_{l_2}^{\hat{k}_{l_2}} \right\}. \quad (6)$$

The optimal solution of  $\mathcal{P}(\phi)$  is given by  $\pi^* = \{\pi^*[1], \dots, \pi^*[\|\mathbb{I}_\phi\|]\}$  and  $\hat{k}_{\pi^*} = \{\hat{k}_{\pi^*[1]}^*, \dots, \hat{k}_{\pi^*[\|\mathbb{I}_\phi\|]}^*\}$ . In (5d) and (5e), the estimated completion time  $EE_l[\hat{k}_l]$  is defined as  $D_l + \varepsilon$  for QoS level 0 and the deadline of QoS level  $\hat{k}_l$  for  $\hat{k}_l \geq 1$ , respectively. Here,  $\varepsilon$  is a constant used to distinguish between  $EE_{\pi[i]}[1]$  and  $EE_{\pi[i]}[0]$ . Note that tasks with QoS level 0 means the rejection of the task, in other words, the task will not be executed.

*Remark 1:* In general, the algorithm (5) is NP hard since finding the set of permutations is NP hard. However, we note

that certain heuristic algorithms such as genetic algorithms [18], [19] may be applied in practice.

*Remark 2:* The obtained plan  $\pi^*$  may not be optimal for the objective function (4) because the estimated completion time for each QoS level ( $\geq 1$ ) is given by the deadline of the corresponding time interval (constraint (5e)). In real-time implementation, a task may be completed before the deadline, which allows for other possibilities of execution. However, we note that the reason for this choice is to guarantee the feasibility of the plan. A way to remedy this is to introduce an online adjustment scheme (e.g., reschedule, restore previous infeasible tasks) at the completion time of each task.

##### B. Controller synthesis

The task execution evolves as follows. Based on the obtained plan  $\pi^*$ , the task  $\pi^*[1]$  will be executed first. Once a task is completed<sup>1</sup>, the agents will proceed immediately to the next one and a new controller will be synthesized and implemented.

Let  $\chi$  be a set that keeps track of the executing tasks. The  $k$ th ( $1 \leq k \leq M$ ) element of  $\chi$  is represented by  $\chi[k]$ , which is given by a triple  $\chi[k] = (E_k, t_0^k, T_k)$ , and  $E_k \in \phi$ ,  $t_0^k$  and  $T_k$  represent the  $k$ th executing task, the starting time of  $E_k$  and the interval of time corresponding to the desired QoS level of  $E_k$ , respectively. For example,  $\chi[2] = (\phi_4, 10, (t_4^2, t_4^1])$  means that the second executing task is  $\phi_4$ , it is executed from time unit 10, and desired to be completed within the time interval  $(t_4^2, t_4^1]$  (QoS level 2). When all tasks are completed, all agents will switch to idle mode, i.e.,  $E_k = T_k = \emptyset$ . Each  $\chi[k]$  determines uniquely a control input  $u_i$  for each agent  $i$ , which is denoted by  $u_i^{\chi[k]}$ .

If  $E_k = \emptyset$  (idle mode), the control input is given by

$$u_i^{\chi[k]} = u_i^{\text{idle}} = -\alpha \sum_{j \in \mathcal{N}_i} (x_i - x_j) - \beta v_i, \quad \text{if } E_k = \emptyset,$$

where  $\alpha, \beta > 0$  are positive control gains. Otherwise, we assume  $(E_k, t_0^k, T_k) = (\phi_l, t_0^k, (t_l^{\hat{k}_l}, t_l^{\hat{k}_l-1}])$ . To guarantee that task  $\phi_l$  is completed at the desired time interval  $(t_l^{\hat{k}_l}, t_l^{\hat{k}_l-1}]$ , we propose an adaptive controller, which is motivated by prescribed performance control [8].

*Definition 1:* A function  $\rho : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{>0}$  will be called a *performance function* if  $\rho$  is continuous, bounded, nonnegative and non-increasing.

*Definition 2:* A function  $S : \mathbb{R} \rightarrow \mathbb{R}$  will be called a *transformation function* if  $S$  is strictly increasing, hence injective and admitting an inverse. In particular, let  $S_1 : [0, 1) \rightarrow \mathbb{R}$  with  $S_1(z) = \ln(\frac{1}{1-z})$ , and  $S_2 : (-1, 1) \rightarrow \mathbb{R}$  with  $S_2(z) = \ln(\frac{1+z}{1-z})$ .

Let  $\mathbf{I}$  be the set of active agents, and  $\mathbf{F} = \mathcal{V} \setminus \mathbf{I}$  be the set of passive agents. For active agent  $i \in \mathbf{I}$ , we propose to prescribe the norm of the tracking error  $\|x_i(t) - c_l\|$  within the following bounds,

$$\alpha_i^k(t) < \|x_i(t) - c_l\| < \beta_i^k(t), \quad i \in \mathbf{I}. \quad (7)$$

<sup>1</sup>A task is said to be completed once all agents lie inside the corresponding target region.

For passive agent  $i \in \mathbf{F}$ , we propose to prescribe the norm of the relative distance between neighboring agents within the following bounds,

$$\|x_{ij}(t)\| < \gamma_i^k(t), \quad i \in \mathcal{V}, (i, j) \in \mathcal{E}, \quad (8)$$

where  $\alpha_i^k(t), \beta_i^k(t), \gamma_i^k(t)$  are performance functions to be defined.

Let  $\underline{t}^k = t_l^{\hat{k}_l}$  and  $\bar{t}^k = t_l^{\hat{k}_l-1}$ . The performance functions  $\alpha_i^k(t), \beta_i^k(t), \gamma_i^k(t)$  are non-increasing. Then, to ensure that  $AE_l \in (\underline{t}^k, \bar{t}^k]$ , it is sufficient to show that the following two conditions,

C1:  $\exists i \in \mathcal{V}$ , such that  $\|x_i(\underline{t}^k) - c_l\| > r_l$ ;

C2:  $\forall i \in \mathcal{V}$ , it holds that  $\|x_i(\bar{t}^k) - c_l\| \leq r_l$ ,

are satisfied simultaneously.

The scheduling algorithm (5) ensures that  $\bar{t}^k > t_0^k$ . However, it is possible that  $\underline{t}^k \leq t_0^k$ . Therefore, in the following, we will present the design of the performance functions and the control synthesis in two different cases, respectively.

1) *Case I:*  $\underline{t}^k \leq t_0^k$ : Define  $\alpha_i^k(t) = 0$  and

$$\beta_i^k(t) = \beta_{i0}^k e^{-\kappa_{i,1}^k(t-t_0^k)}, \quad (9a)$$

$$\gamma_i^k(t) = \gamma_{i0}^k e^{-\mu_{i,1}^k(t-t_0^k)}, \quad (9b)$$

for  $t \geq t_0^k$ , where  $\beta_{i0}^k > \max\{\|x_i(t_0^k) - c_l\|, \sigma_k r_l\}$  and  $\gamma_{i0}^k > \max\{\max_{j \in \mathcal{N}_i} \{\|x_{ij}(t_0^k)\|\}, r_{\min}/(N-1)\}$ . In addition,

$$\kappa_{i,1}^k = \frac{1}{(\bar{t}^k - t_0^k)} \ln \frac{\beta_{i0}^k}{\sigma_k r_l}, \quad (10a)$$

$$\mu_{i,1}^k = \frac{1}{(\bar{t}^k - t_0^k)} \ln \frac{(N-1)\gamma_{i0}^k}{(1-\sigma_k)r_{\min}}, \quad (10b)$$

where  $\sigma_k \in (0, 1)$ .

*Remark 3:* The definitions of  $\beta_{i0}^k, \gamma_{i0}^k$  guarantee that the performance bounds (7) and (8) are satisfied at starting time  $t_0^k$ . In addition, from (9a) and (10a), one can get

$$\|x_i(\bar{t}^k) - c_l\| < \beta_i^k(\bar{t}^k) = \beta_{i0}^k e^{-\kappa_{i,1}^k(\bar{t}^k - t_0^k)} = \sigma_k r_l, \forall i \in \mathbf{I}.$$

Moreover, from (9b) and (10b), one can get

$$\begin{aligned} \|x_{ij}(\bar{t}^k)\| &< \gamma_j^k(\bar{t}^k) = \gamma_{j0}^k e^{-\mu_{j,1}^k(\bar{t}^k - t_0^k)} \\ &= (1-\sigma_k)r_{\min}/(N-1), \forall (i, j) \in \mathcal{E}. \end{aligned}$$

Since the graph  $\mathcal{G}$  is connected, there are at most  $N-1$  edges between a passive agent and an active agent, so then one has  $\|x_i(\bar{t}^k) - c_l\| < (N-1)(1-\sigma_k)r_{\min}/(N-1) + \sigma_k r_l \leq r_l, \forall i \in \mathbf{F}$ . That is, C2 is satisfied.

Based on Remark 3, one can conclude that if for task  $E_k$ , i) the tracking error  $\|x_i - c_l\|, \forall i \in \mathbf{I}$  is evolving within the prescribed performance bound (7), and ii) the relative distance  $\|x_{ij}\|, \forall (i, j) \in \mathcal{E}$  is evolving within the prescribed performance bound (8) for  $t \geq t_0^k$ , then the task  $E_k$  will be completed within the desired time interval  $(\underline{t}^k, \bar{t}^k]$ .

The approach can be explained as follows. To guarantee that C2 is satisfied, we first ensure that the active agents  $i \in \mathbf{I}$  will reach the ball area around the target point  $c_l$  with radius  $\sigma_k r_l$ , that is,

$$\mathbb{X}_l^{\mathbf{I}} := \{z \in \mathbb{R}^n : \|z - c_l\| < \sigma_k r_l\}, \quad (11)$$

before  $\bar{t}^k$ . Then, we further ensure that the relative distance between neighboring agents will reach

$$\mathbb{X}_l^{\mathcal{E}} := \{z \in \mathbb{R}^n : \|z\| < \frac{(1-\sigma_k)r_{\min}}{N-1}\}, \quad (12)$$

before  $\bar{t}^k$ . As stated in Remark 3, (11) and (12) together imply that C2 is satisfied.

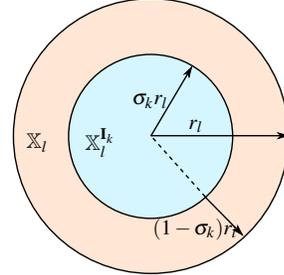


Fig. 2. Target regions for active agents and passive agents ( $n = 2$ )

*Remark 4:* We assume that the agents have no information about the communication graph  $\mathcal{G}$  (otherwise, a linear feedback controller can be designed, which is similar to [13]). Furthermore, the passive agents do not know if they are connected to active agents. Therefore, in (12), we set the prescribed relative distance between neighboring agents to be  $(1-\sigma_k)r_{\min}/(N-1)$ , which corresponds to the ‘worst-case’ scenario, at the expense of knowing the total number of agents  $N$ . This level of centralized knowledge can be attained offline and is typical in many multi-agent approaches for less complex problems than the one in hand [20].

Define the normalized errors as

$$\xi_i(t) = \frac{\|x_i(t) - c_l\|}{\beta_i^k(t)}, \quad \xi_{ij}(t) = \frac{\|x_{ij}(t)\|}{\gamma_j^k(t)}, \quad (13)$$

respectively. Now, (7) is equivalent to  $0 < \xi_i(t) < 1$  and (8) is equivalent to  $0 \leq \xi_{ij}(t) < 1$ .

The normalized errors  $\xi_i$  and  $\xi_{ij}$  are transformed through transformation function  $S_1$ . We denote the transformed error  $\zeta_i(\xi_i)$  and  $\varepsilon_{ij}(\xi_{ij})$  by  $\zeta_i(\xi_i) = S_1(\xi_i), \varepsilon_{ij}(\xi_{ij}) = S_1(\xi_{ij})$ , respectively, where we dropped the time argument  $t$  for notation convenience.

Let  $\nabla S_1(\xi_i) = \partial S_1(\xi_i)/\partial \xi_i, \nabla S_1(\xi_{ij}) = \partial S_1(\xi_{ij})/\partial \xi_{ij}$ . We propose the following time-varying control protocol:

$$\begin{aligned} u_i^{\chi[k]} &= - \sum_{j \in \mathcal{N}_i} \frac{1}{\gamma_j^k} \nabla S_1(\xi_{ij}) \varepsilon_{ij}(\xi_{ij}) \mathbf{n}_{ij} \\ &\quad - \frac{h_i^k}{\beta_i^k} \nabla S_1(\xi_i) \zeta_i(\xi_i) \mathbf{n}_i - K_i^k v_i, \end{aligned} \quad (14)$$

where  $h_i^k = 1, i \in \mathbf{I}$  and  $h_i^k = 0, i \in \mathbf{F}$ ,  $\mathbf{n}_i = (x_i - c_l)/\|x_i - c_l\|, \mathbf{n}_{ij} = x_{ij}/\|x_{ij}\|$ , and  $K_i^k$  is a positive control gain to be determined later.

*Theorem 1:* Consider the MAS (1) and the prescribed performance controller for task  $\chi[k]$  given by (14) with  $\underline{t}^k \leq t_0^k$ . Suppose Assumption 1 holds and the control gain  $K_i^k$  satisfies  $K_i^k > \max\{\mu_{i,1}^k, \kappa_{i,1}^k\}, \forall i$ . Then, i) the tracking error

$\|x_i - c_l\|, \forall i \in \mathbf{I}$  will evolve within the performance bound (7), ii) the relative distance  $\|x_{ij}\|, (i, j) \in \mathcal{E}$  will evolve within the performance bound (8) for  $t \geq t_0^k$ , iii) the control signal (14) is bounded for a finite completion time.

The proof is omitted due to space limitations. For details, please refer to the extended version [21].

2) *Case II:*  $t^k > t_0^k$ : The performance functions  $\alpha_i^k, \beta_i^k, \gamma_i^k$  are defined as

$$\alpha_i^k(t) = \begin{cases} \alpha_{i0}^k e^{-\kappa_{i,2}^k(t-t_0^k)}, & t \in [t_0^k, t^k] \\ r_l e^{-\kappa_{i,3}^k(t-t^k)}, & t > t^k, \end{cases} \quad (15a)$$

$$\beta_i^k(t) = \begin{cases} \beta_{i0}^k e^{-\kappa_{i,2}^k(t-t_0^k)}, & t \in [t_0^k, t^k] \\ \frac{\beta_{i0}^k r_l}{\alpha_{i0}^k} e^{-\kappa_{i,3}^k(t-t^k)}, & t > t^k, \end{cases} \quad (15b)$$

$$\gamma_i^k(t) = \begin{cases} \gamma_{i0}^k e^{-\kappa_{i,2}^k(t-t_0^k)}, & t \in [t_0^k, t^k] \\ \frac{\gamma_{i0}^k r_l}{\alpha_{i0}^k} e^{-\mu_{i,2}^k(t-t^k)}, & t > t^k, \end{cases} \quad (15c)$$

where

$$\alpha_{i0}^k = \|x_i(t_0^k) - c_l\| - \Delta_i^k, \quad (16a)$$

$$\beta_{i0}^k = \|x_i(t_0^k) - c_l\| + \Delta_i^k, \quad (16b)$$

$$\gamma_{i0}^k > \max \left\{ \max_{j \in \mathcal{N}_i} \{\|x_{ij}(t_0^k)\|\}, \frac{r_{\min}}{N-1} \right\}, \quad (16c)$$

and  $0 < \Delta_i^k < \|x_i(t_0^k) - c_l\| - r_l, \forall i \in \mathbf{I}$ . In addition,

$$\kappa_{i,2}^k = \frac{1}{(t^k - t_0^k)} \ln \frac{\alpha_{i0}^k}{r_l}, \quad (17a)$$

$$\kappa_{i,3}^k = \frac{1}{(t^k - t^k)} \ln \frac{\beta_{i0}^k}{\sigma_k \alpha_{i0}^k}, \quad (17b)$$

$$\mu_{i,2}^k = \frac{1}{(t^k - t^k)} \ln \frac{(N-1)\gamma_{i0}^k}{(1-\sigma_k)r_{\min}}, \quad (17c)$$

where  $\sigma_k \in (0, 1)$  is a constant to be determined later. One can verify that the functions  $\alpha_i^k, \beta_i^k, \gamma_i^k$  satisfy the definition of performance function.

*Proposition 1:* The performance functions  $\alpha_i^k, \beta_i^k$  and  $\gamma_i^k$  defined in (15a), (15b) and (15c) guarantee that the conditions C1 and C2 are satisfied simultaneously.

Let  $\rho_i^k(t) \triangleq (\beta_i^k(t) + \alpha_i^k(t))/2$ ,  $\delta_i^k(t) \triangleq (\beta_i^k(t) - \alpha_i^k(t))/2$ . Then, (7) can be rewritten as  $-\delta_i^k(t) + \rho_i^k(t) < \|x_i(t) - c_l\| < \rho_i^k(t) + \delta_i^k(t)$ . Define in this case the normalized error  $\xi_i(t)$  as  $\xi_i(t) = (\|x_i(t) - c_l\| - \rho_i^k(t))/\delta_i^k(t)$ , and  $\xi_{ij}$  is defined the same as in (13).

The normalized errors  $\xi_i$  and  $\xi_{ij}$  are transformed through transformation functions  $S_2$  and  $S_1$ , respectively. We denote the transformed errors  $\zeta_i(\xi_i)$  and  $\varepsilon_{ij}(\xi_{ij})$  by  $\zeta_i(\xi_i) = S_2(\xi_i)$ ,  $\varepsilon_{ij}(\xi_{ij}) = S_1(\xi_{ij})$ .

Let  $\nabla S_2(\xi_i) = \partial S_2(\xi_i)/\partial \xi_i$ ,  $\nabla S_1(\xi_{ij}) = \partial S_1(\xi_{ij})/\partial \xi_{ij}$ . We propose the following time-varying control protocol:

$$u_i^{X[k]} = - \sum_{j \in \mathcal{N}_i} \frac{1}{\gamma_i^k} \nabla S_1(\xi_{ij}) \varepsilon_{ij}(\xi_{ij}) \mathbf{n}_{ij} - \frac{h_i^k}{\delta_i^k} \nabla S_2(\xi_i) \zeta_i(\xi_i) \mathbf{n}_i - K_i^k v_i. \quad (18)$$

Then, the following holds.

*Theorem 2:* Consider the MAS (1) and the prescribed performance controller for task  $\chi[k]$  given by (18) with  $t^k > t_0^k$ . Suppose Assumption 1 holds and the constants  $\sigma_k, K_i^k$  satisfy

$$\sigma_k \leq \frac{r_{\min}}{(N-1)\bar{\gamma}^k + r_{\min}}, \quad K_i^k > \max \left\{ \kappa_{i,2}^k, \kappa_{i,3}^k \right\}, \forall i, \quad (19)$$

where  $\bar{\gamma}^k \geq \max_{i \in \mathcal{V}} \{\gamma_{i0}^k\}$ . Then, i) the tracking error  $\|x_i - c_l\|, \forall i \in \mathbf{I}$  will evolve within the performance bound (7), ii) the relative distance  $\|x_{ij}\|, (i, j) \in \mathcal{E}$  will evolve within the performance bound (8) for  $t \geq t_0^k$ , iii) the control signal (18) is bounded for a finite completion time.

The proof is omitted due to space limitations. For details, please refer to the extended version [21].

*Remark 5:* We note that if for task  $\phi_l$ , one has  $\mathbf{I}_l = \mathcal{V}$ , i.e., all agents are active, then it is possible to design a linear feedback controller for each agent  $i$ , such that the task  $\phi_l$  is completed at any desired QoS level.

*Remark 6:* In practice, it may not be possible to use prescribed performance controller to drive an agent to a given target region within any time interval due to physical constraints (e.g., input constraints). However, given the initial conditions and the upper bound of the prescribed performance controller, one may able to calculate (or estimate) the minimal time required to drive one agent to a given target region (e.g., using reachability analysis). Then, the constraint (5f) in (5) can be revised as

$$EE_{\pi[i+1]}[\hat{k}_{\pi[i+1]}] \geq EE_{\pi[i]}[\hat{k}_{\pi[i]}] + t_{\min}(\mathbb{X}_{\pi[i]}, \mathbb{X}_{\pi[i+1]}),$$

where  $t_{\min}(\mathbb{X}_{\pi[i]}, \mathbb{X}_{\pi[i+1]})$  represents the minimal time required to navigate from region  $\mathbb{X}_{\pi[i]}$  to  $\mathbb{X}_{\pi[i+1]}$ .

## V. SIMULATION

In this section, a numerical example is given to verify the theoretical results. Consider a network of 4 agents with  $n=2$ , and the communication graph  $\mathcal{G}$  is shown in Fig. 3, where agent 1 is active and agents 2,3,4 are passive. The initial position  $x_i(0)$  of each agent  $i$  is chosen randomly from the box  $[0, 2] \times [0, 2]$ , and the initial velocity  $v_i(0)$  of each agent  $i$  is chosen to be  $[0, 0]^T, \forall i$ .



Fig. 3. Communication graph among the agents.

The high level plan  $\phi$  consists of 3 tasks  $\phi_1, \phi_2, \phi_3$ , the corresponding target sets are given by:  $\mathbb{X}_1 = \mathcal{B}([10, 8]^T, 1)$  with deadline  $D_1 = 15$ ;  $\mathbb{X}_2 = \mathcal{B}([3, 10]^T, 1)$  with deadline  $D_2 = 20$ ;  $\mathbb{X}_3 = \mathcal{B}([5, 5]^T, 1)$  with deadline  $D_3 = 24$ . Tasks  $\phi_1, \phi_3$  have 4 QoS levels, while task  $\phi_2$  has 2 QoS levels, respectively. The corresponding time intervals and rewards are given by

$$\begin{aligned} \phi_1 &: \{(15, +\infty), -20\}, \{(10, 15], 5\}, \{(5, 10], 10\}, \{(0, 5], 8\}; \\ \phi_2 &: \{(20, +\infty), -20\}, \{(0, 20], 10\}; \\ \phi_3 &: \{(24, +\infty), -20\}, \{(14, 24], 5\}, \{(9, 14], 10\}, \{(0, 9], 5\}. \end{aligned}$$

The objective function is given in (4) with  $\alpha = 0.8$ . Then by solving (5), one can get that the optimal solution is  $\{(\pi^*[1], \hat{k}_{\pi^*[1]}^*), (\pi^*[2], \hat{k}_{\pi^*[2]}^*), (\pi^*[3], \hat{k}_{\pi^*[3]}^*)\} = \{(1, 3), (3, 3), (2, 2)\}$ .

The simulation results are shown in Figs. 4-6. Fig.4 shows the evolution of positions for each agent. The three red circles (from left to right) represent the three target regions  $\mathbb{X}_1, \mathbb{X}_3$  and  $\mathbb{X}_2$ , respectively. The evolution of tracking error  $\|x_1 - c_l\|$  for active agent 1 and the performance bounds  $\alpha_1^k, \beta_1^k$  are depicted in Fig. 5. In addition, the evolution of relative distances between neighboring agents and the performance bounds  $\gamma_{1,2,3,4}^k$  are plotted in Fig. 6. One can see that the performance bounds are satisfied at all times.

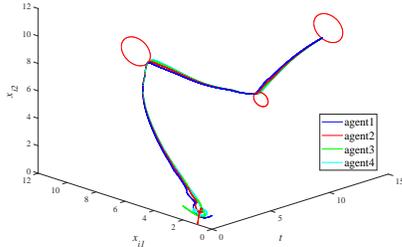


Fig. 4. The evolution of positions for each agent under (14).

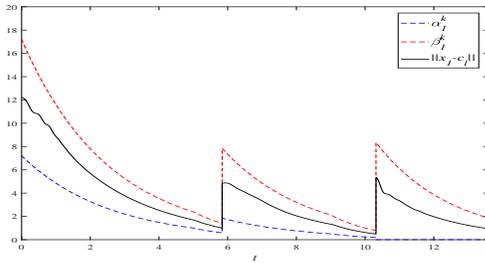


Fig. 5. The evolution of the tracking error for active agent 1 and performance bounds.

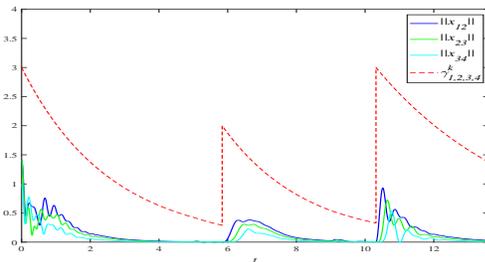


Fig. 6. The evolution of the relative distances between neighboring agents and performance bounds.

## VI. CONCLUSIONS

We proposed a task scheduling algorithm and distributed control design for the coordination of MAS that is requested

to visit a sequence of target regions with deadline constraints. By utilizing ideas from prescribed performance control, we developed a hybrid feedback control law that guarantees the satisfaction of each task under specific time interval constraints. A natural next step is to consider more complex task specifications and perform physical experiments.

## REFERENCES

- [1] M. Ji and M. B. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness", *IEEE Trans. Robot.*, Vol. 23, pp. 693-703, 2007.
- [2] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies", *IEEE Trans. Autom. Control*, Vol. 50, pp. 655-661, 2005.
- [3] 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, pp. 820-826, 2014.
- [4] K. Ramamritham and J. A. Stankovic, "Dynamic task scheduling in hard real-time distributed systems", *IEEE software*, Vol. 1, pp. 65-75, 1984.
- [5] C. L. Liu and James W. Layland, "Scheduling algorithms for multi-programming in a hard-real-time environment", *Journal of the ACM (JACM)*, Vol. 20, pp. 46-61, 1973.
- [6] M. Klein, T. Ralya, B. Pollak, R. Obenza and M. G. Harbour, A practitioner's handbook for real-time analysis: guide to rate monotonic analysis for real-time systems, Springer Science & Business Media, 2012.
- [7] J. A. Stankovic, M. Spuri, K. Ramamritham and G. C. Buttazzo. Deadline scheduling for real-time systems: EDF and related algorithms, Springer Science & Business Media, 2012.
- [8] C. P. Bechlioulis and G. A. Rovithakis, "Robust adaptive control of feedback linearizable MIMO nonlinear systems with prescribed performance", *IEEE Trans. Autom. Control*, Vol. 53, pp. 2090-2099, 2008.
- [9] Y. Karayiannidis, D. V. Dimarogonas and D. Kragic, "Multi-agent average consensus control with prescribed performance guarantees", in *Proc. Decision and Control (CDC)*, pp. 2219-2225, 2012.
- [10] L. Macellari, Y. Karayiannidis, D. V. Dimarogonas, "Multi-agent second order average consensus with prescribed transient behavior", *IEEE Trans. Autom. Control*, Vol. 62, pp. 5282-5288, 2017.
- [11] C. P. Bechlioulis, M. A. Demetriou and K. J. Kyriakopoulos, "A distributed control and parameter estimation protocol with prescribed performance for homogeneous lagrangian multi-agent systems", *Autonomous Robots*, pp. 1-17, 2018.
- [12] C. K. Verginis, C. P. Bechlioulis, D. V. Dimarogonas and K. J. Kyriakopoulos, "Robust distributed control protocols for large vehicular platoons with prescribed transient and steady-state performance", *IEEE Trans. Control Syst. Technol.*, Vol. 26, pp. 299-304, 2018.
- [13] M. Guinaldo and D. V. Dimarogonas, "A hybrid systems framework for multi agent task planning and control", in *Proc. American Control Conference (ACC)*, pp. 1181-1186, 2017.
- [14] L. Lindemann, C. K. Verginis and D. V. Dimarogonas, "Prescribed performance control for signal temporal logic specifications", in *Proc. Decision and Control (CDC)*, pp. 2997-3002, 2017.
- [15] T. Abdelzaher and K. G. Shin, "End-host architecture for QoS-adaptive communication", in *Proc. Real-Time Technology and Applications Symposium*, pp. 121-130, 1998.
- [16] C. Lu, J. A. Stankovic, S. H. Son and G. Tao, "Feedback control real-time scheduling: Framework, modeling, and algorithms", *Real-Time Systems*, Vol. 23, pp. 85-126, 2002.
- [17] M. Guo, J. Tumova and D. V. Dimarogonas, "Communication-free multi-agent control under local temporal tasks and relative-distance constraints", *IEEE Trans. Autom. Control*, Vol. 61, pp. 3948-3962, 2016.
- [18] P. T. Zacharia and N. A. Aspragathos, "Optimal robot task scheduling based on genetic algorithms", *Rob. Comput. Integr. Manuf.*, Vol. 21, pp. 67-79, 2005.
- [19] F. A. Omara and M. M. Arafa, "Genetic algorithms for task scheduling problem", *J. Parallel Distrib. Comput.*, Vol. 70, pp. 13-22, 2010.
- [20] M. Mesbahi and M. Egerstedt, "Graph theoretic methods in multiagent networks", *Princeton University Press*, 2010.
- [21] P. Yu, and D. V. Dimarogonas, "Time-constrained multi-agent task scheduling based on prescribed performance control", arXiv:1804.10018 [cs.SY].