Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Event-Triggered Task-Switching Control Based on Distributed Estimation

Xingkang He * **Ehsan Hashemi** ** **Karl H. Johansson** *

* *Division of Decision and Control Systems, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden ((xingkang,kallej)@kth.se)*
** *Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, ON, Canada (ehashemi@uwaterloo.ca)*

Abstract: This paper studies how to control an agent in an uncertain environment over a connected sensor network, such that the agent is able to finish a sequence of tasks, namely, reaching certain sets in order. Based on multiple offline reference trajectories and constrained communication between the agent and the sensor network, an event-triggered task-switching control framework is proposed, so that the agent state remains in each task set for the desired time and then switches to the next task. Employing a local predicted control law and the messages from neighboring sensors, a two time-scale distributed filter is proposed for each sensor to estimate the agent state. Under mild system conditions (i.e., stabilization and collective detectability), the estimation error and trajectory tracking error are shown to be asymptotically upper bounded.

Keywords: Task-switched, distributed estimation, event-triggered, sensor network, trajectory tracking

1. INTRODUCTION

The well-known separation principle, where the optimal controller and state estimator for linear systems can be designed separately, depends on the condition that the controller and estimator share the full state and decision-making knowledge to each other all the time. Regarding the networked control systems, such as cyber-physical systems, mobile multi-robot systems, and networked vehicular systems, to avoid heavy channel burden and energy consumption induced by tremendous communication between sensors and the control center, an intuitive way to keep the separation principle is choosing a small set of sensors which share the information (measurements or locally estimated values) with the control center timely. However, the information of these sensors may not be sufficient to provide satisfactory state estimates, which then hinders the performance of estimate-based controllers.

To deal with the problems where the separation principle does not hold, distributed estimation and control have been widely utilized to improve reliability and convergence behavior of multi-agent and networked robotic systems (Yang et al. (2008); Wang and Gu (2011); Li et al. (2014)). A simultaneous collective localization approach for a mobile multi-robot network, where each robot shares its local sensory data to others, is proposed in Roumeliotis and Bekey (2002) by using a decentralized Kalman filter. Nonlinear Lyapunov-based tracking control is combined with adaptive switching supervisory control in Aguiar and Hespanha (2007) to improve convergence of the trajectory tracking error and to bound it to an arbitrary small neighborhood of an origin. A distributed observer and controller framework is designed in Antonelli et al. (2014) for a wheeled multi-robot system and its performance is studied for strongly connected switching and non-switching topologies. In order to estimate inertial and kinematic parameters of an unknown rigid body, a distributed filter is designed for a network of mobile robots in Franchi et al. (2015), in which the manipulation strategy is functional to the estimation process to satisfy observability conditions. In Freundlich et al. (2017) an information consensus filter is used to reconfigure a network of mobile sensors for estimating a set of hidden states up to a user-defined accuracy and to design a controller robust to the state disagreement errors. In Miao et al. (2018), a distributed formation control law is designed based on the estimated states of the leader agent, and asymptotic convergence of formation tracking errors is studied under some mild assumptions on the interaction graph among the leader and the follower agents. A distributed observer is designed in Marino (2017) for state estimation in a connected cooperative robot framework, in which the estimates are used to calculate an adaptive local control law to deal with model uncertainties. However, most of the above results are given for the control and estimation of collective behaviors of sparsely connected multi agents, which may not work well for centralized control problems involving complex decision-makings or the systems with limited control channels.

In this paper, under constrained communications between an agent and a connected sensor network, an event-triggered task-switching framework [1] based on distributed estimation is proposed to control an agent following preset multiple trajectories for reaching a sequence of task sets (e.g., the coverage of a wide area). We propose a two time-scale distributed filter for the estimation of agent state based on a local predicted control law and the communication of neighboring sensors. We provide the conditions such that the state estimation error and the trajectory tracking error are asymptotically upper bounded.

The remainder of the paper is organized as follows. Section 2 is on problem formulation. Section 3 studies a task-switching

⋆ The work is supported by Knut & Alice Wallenberg foundation, and by Swedish Research Council.

[1] The task-switching strategy is to adjust the current task if certain condition or objective is met.

control framework. Section 4 analyzes the boundedness of the estimation error and trajectory tracking error. Numerical simulations are given in Section 5. Section 6 concludes this paper. Due to page limitation, the proofs are omitted.

Notations. $\mathrm{diag}\{\cdot\}$ means that elements are arranged in diagonals. $\mathbf{1}_N$ stands for the N-dimensional vector with all elements being one. $A \otimes B$ is the Kronecker product of A and B. $\|x\|$ is the 2-norm of a vector x. $\|A\|$ is the induced 2-norm, i.e., $\|A\| = \sup_{x\neq 0} \frac{\|Ax\|}{\|x\|}$. $\lambda_{min}(A)$, $\lambda_2(A)$ and $\lambda_{max}(A)$ are the minimal, second minimal and maximal eigenvalues of a real-valued symmetric A, respectively. $\lceil \cdot \rceil$ is the ceiling operation. $\mathcal{I}_{cond}$ is a $0-1$ indicator, where $\mathcal{I}_{cond} = 1$ if $cond$ holds, otherwise $\mathcal{I}_{cond} = 0$. All the matrices and vectors in this paper are real-valued.

2. PROBLEM FORMULATION

2.1 System model

Consider a discrete-time agent state observed by N sensors

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + w(t) \\ y_i(t) &= C_i x(t) + v_i(t), i = 1, \ldots, N, \end{aligned} \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the agent state, $u(t) \in \mathbb{R}^p$ the control input, $w(t) \in \mathbb{R}^n$ the process noise, $y_i(t) \in \mathbb{R}^{M_i}$ the observation vector of sensor i, and $v_i(t) \in \mathbb{R}^{M_i}$ the observation noise. The real-valued matrices A, B, C_i have compatible dimensions.

In this work, we consider a distributed communication scheme for a sensor network without self-loop, where each sensor simply shares information with its neighboring sensors. We model the communication topology of these sensors through a fixed undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V} = \{1, 2, \ldots, N\}$ denotes the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ the set of edges, and $\mathcal{A}$ the 0-1 adjacency matrix. If the (i, j)th element of $\mathcal{A}$ is 0, there is an edge $(i, j) \in \mathcal{E}$, through which node i can exchange information with node j. In the case, node j is called a neighbor of node i, and vice versa. Let the neighbor set of node i be $\mathcal{N}_i := \{j \in \mathcal{V} | (i, j) \in \mathcal{E}\}$. Suppose that $\mathcal{D}$ is the degree matrix, which is a diagonal matrix consisting of the numbers of neighbors. Denote $\mathcal{L} = \mathcal{D} - \mathcal{A}$ the Laplacian matrix. The graph $\mathcal{G}$ is connected if for any pair of nodes (i_1, i_l), there exists a path from i_1 to i_l consisting of edges $(i_1, i_2), (i_2, i_3), \ldots, (i_{l-1}, i_l)$. It is known that graph $\mathcal{G}$ is connected if and only if $\lambda_2(\mathcal{L}) > 0$.

In this paper, the following assumption is needed.

***Assumption** 2.1.* The following conditions hold:

(i) There are bounded scalars q_w, q_v and q_x known to the control center (i.e., agent) such that for any $i \in \mathcal{V}$,
$$\|w(t)\| \leq q_w, \|v_i(t)\| \leq q_v, \|\hat{x}_i(0) - x(0)\| \leq q_x,$$
where $\hat{x}_i(0)$ is the initial estimate of sensor i.

(ii) The system is stabilizable with $A + BK$ is Schur stable, where K is the control gain matrix utilized in equation (5).

(iii) The system is collectively detectable such that $A - GC$ is Schur stable, where
$$\begin{aligned} G &= \frac{1}{N}(G_1, G_2, \ldots, G_N), \\ C &= (C_1^T, C_2^T, \ldots, C_N^T)^T, \end{aligned} \tag{2}$$
and G_i is the filtering gain utilized in Algorithm 2.

(iv) The graph $\mathcal{G}$ is undirected and connected.

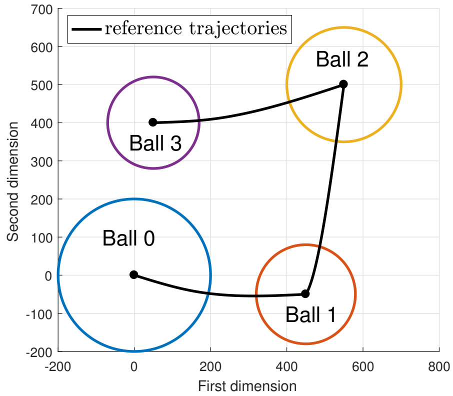


Fig. 1. An example with three task balls (i.e., balls 1–3) and reference trajectories for a two-dimensional system. Ball 0 contains the initial system state. The agent state is controlled to reach the task balls in order by following the reference trajectories.

The stabilization in (ii) provides the design principle of static control gain. The collective observability in (iii) is the mildest condition in distributed estimation. The connectivity in (iv) ensures the information flow over the sensor network. Under the condition of weak observability, e.g., (A, C_s) is not observable, the centralized controller can not simply utilize the information of an isolated sensor[2] s to stabilize the system (1). Thus, it is necessary to design a collaborating scheme for these sensors to obtain effective state estimates, which then enable the control center to design state estimates based controller for a certain objective. In this paper, we model the task sets by multi-dimensional balls defined as follows.

***Definition** 2.1.* The set

$$\mathbb{O}(c_i, R_i) = \{x \in \mathbb{R}^n | \, \|x - c_i\| \leq R_i\} \tag{3}$$

is called a **ball** with center $c_i \in \mathbb{R}^n$ and radius $R_i \in \mathbb{R}$.

Motivated by the example that a robot is desired to cover a wide area by reaching certain places, we aim to design a control strategy such that the system state can be driven into k_* balls (i.e., $\{\mathbb{O}(c_i, R_i)\}_{i=1}^{k_*}$) one after another. We also call the k_* balls as task balls in the sequel. Without losing generality, we suppose the label sequence of the k_* task balls is in order, i.e., $\{1, 2, \ldots, k_*\}$. Moreover, we assume there is an initial error ball $\mathbb{O}(c_0, R_0)$, i.e., $x(0) \in \mathbb{O}(c_0, R_0)$. Assume there is a reference trajectory $\{r_{i,i+1}(k)\}_{k=1}^{T_{i,i+1}}$ connecting the centers of two adjacent balls $\mathbb{O}(c_i, R_i)$ and $\mathbb{O}(c_{i+1}, R_{i+1})$, where $T_{i,i+1}$ is the data length of the reference trajectory from balls i to $i+1$, where $i = 0, 1 \ldots, k_* - 1$. On the reference trajectory, the following assumption is in need.

***Assumption** 2.2.* For any $i = 0, 1, \ldots, k_* - 1$, it holds that for $k = 1, \ldots, T_{i,i+1} - 1$,

$$\begin{aligned} r_{i,i+1}(1) &= c_i, \\ r_{i,i+1}(k+1) &= Ar_{i,i+1}(k) + Bu_{i,i+1}(k), \\ r_{i,i+1}(T_{i,i+1}) &= c_{i+1}, \end{aligned} \tag{4}$$

where $\{r_{i,i+1}(k)\}_{k=1}^{T_{i,i+1}}$ and $\{u_{i,i+1}(k)\}_{k=1}^{T_{i,i+1}}$ are offline stored at the control center and all sensors. Moreover, $\|r_{0,1}(1) - x(0)\|\} \leq q_r$.

[2] Here, an isolated sensor means that the sensor does not communicate with other sensors, and it obtains the state estimates based on its own observations.

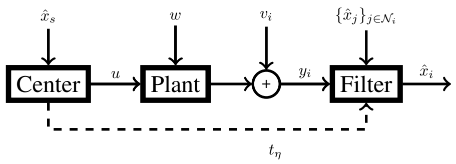


Fig. 2. Each sensor is equipped with a distributed filter providing an estimate $\hat{x}_i$ of the plant x. The center receives the estimate $\hat{x}_s$ from sensor s and generates the control u. If the event-triggered conditions in Algorithm 1 are met, the center will broadcast the task-switching time t_η to all sensors.

Assumption 2.2 indicates that there are k_* reference trajectories, which starts from the center of the initial error ball and connects the centers of the k_* task balls in order. An example is provided in Fig. 1.

3. TASK-SWITCHED CONTROL FRAMEWORK

In this section, we consider a framework of event-triggered task-switching control based on distributed state estimation. First, we study how to design an event-triggered task-switching control and then propose a distributed filter to estimate the controlled agent state.

3.1 Task-switched control

Let T_u stands for the initial control time, which means the control input before T_u is zero, i.e., $u(t) = 0$, if $t < T_u$. Based on the estimate of sensor $s(t)$, $s(t) \in \mathcal{V}$, and the reference trajectory in (4), the control input $u(t)$ for the system (1) is designed as follows: for $t \geq T_u$,

$$\begin{aligned} u(t) &= K\left(\hat{x}_{s(t)}(t) - r(t)\right) + u_r(t), \\ r(t) &= r_{\eta,\eta+1}(t - t_\eta + 1), \\ u_r(t) &= u_{\eta,\eta+1}(t - t_\eta + 1), \end{aligned} \tag{5}$$

where $\hat{x}_{s(t)}(t)$ is the estimate for $x(t)$ by sensor $s(t) \in \mathcal{V}$, $K \in \mathbb{R}^{p\times n}$ the static control gain of state feedback satisfying the requirement in Assumption 2.1. Since the reference trajectories are in order, given a sequence of t_η, η is able to be updated correspondingly. In other words, $r(t)$ and $u_r(t)$ in (5) are determined by simply sharing t_η between the center and sensors. Then, a diagram on the whole framework is provided in Fig. 2.

Communication of the framework in Fig. 2: 1) When the event-triggered conditions are met, the control center will broadcast a task-switching time t_η, after which the task label is switched from η to $\eta + 1$. In other words, t_η is the start time for the task ball $\eta + 1$; 2) The control center can access the state estimates of one sensor, whose label could be time-varying. An example is that a controlled moving vehicle can choose the nearest sensor to obtain its current state estimate. With the moving of the vehicle, the relative positions between the sensors and the vehicle are changing. 3) The sensors can communicate with each other in a distributed manner over the sensor network.

Some remarks are given: (a) Although a task-switching time t_η is broadcast by the center to all the sensors, the exact control signal $u(t)$ in equation (5) is unknown to all sensors but sensor $s(t)$. It is different from the classical centralized control, where the exact control signal is shared between each sensor and the control center. (b) Different from the traditional centralized estimation framework (e.g., centralized Kalman filter) where all the observations of sensors need to be transmitted to the center, the proposed framework only requires that one sensor (i.e., sensor $s(t) \in \mathcal{V}$) transmits its state estimates to the center.

Algorithm 1 Event-triggered task-switching control

Initial setting: Control gain K satisfying Assumption 2.1, initial control time $T_u \geq 0$, task balls $\{\mathbb{O}(c_j, R_j)\}_{j=1}^{k_*}$, reference trajectories $\{r_{i,i+1}(k)\}_{k=1}^{T_{i,i+1}}$ and inputs $\{u_{i,i+1}(k)\}_{k=1}^{T_{i,i+1}}$ satisfying Assumption 2.2, $i = 0, 1, \ldots, k_* - 1$, desired time length inside task balls $\{\mathbb{T}_i\}$, cumulated remaining time set $\{\mathcal{T}_{i+1}\}$ initialized by empty sets, initial task-switching time $t_0 = T_u$ and $\eta = 0$;
For $t = T_u, T_u + 1, \ldots$
  The control center receives $\hat{x}_{s(t)}(t)$ from sensor $s(t)$
  Control law:
    $u(t)$ is given in (5);
  Event-triggered switching scheme: The control center calculates $f(t)$ with $\chi(t)$ and $\varphi(t)$ both in (6)::

$$\begin{aligned} f(t) = \min\big\{&\chi(t) + \left\|\hat{x}_{s(t)}(t) - c_{\eta+1}\right\|, \\ &\varphi(t) + \|r(t) - c_{\eta+1}\|\,\big\} \end{aligned}$$

  If $f(t) \leq R_\eta$:
    If $\mathcal{T}_i = \emptyset$ or $t - 1 \notin \mathcal{T}_i$
      $\mathcal{T}_i = \{t\}$
    Else
      $\mathcal{T}_i = \{t\} = \mathcal{T}_i \bigcup \{t\}$
    EndIf
    If $|\mathcal{T}_i| \geq \mathbb{T}_i$
      $\eta = \eta + 1, t_\eta = t + 1,$
      Broadcast t_η to all sensors;
    EndIf
  ElseIf $t = T_{\eta,\eta+1} + t_\eta - 1$:
    $\eta = \eta + 1, t_\eta = t + 1,$
    Broadcast t_η to all sensors;
  EndIf
EndFor

***Definition** 3.1.* The set

$$\mathbb{O}(\hat{x}(t), \tilde{r}(t)) = \{x \in \mathbb{R}^n \,|\, \|x - \hat{x}(t)\| \leq \tilde{r}(t)\}$$

is called a **controlled ball** at time t with center $\hat{x}(t) \in \mathbb{R}^n$ and radius $\tilde{r}(t) \in \mathbb{R}$.

Since the system state $x(t)$ in (1) is unknown, to judge whether it stays in the task ball, we provide two methods The first method is based on Definition 3.1. The set $\mathbb{O}(\hat{x}_{s(t)}(t), \chi(t))$ is controlled ball, if there is a sequence $\{\chi(t)\}$ so that $\left\|x(t) - \hat{x}_{s(t)}\right\| \leq \chi(t)$. Such $\chi(t)$ will be provided in Theorem 4.1. Then, the state $x(t)$ will definitely go into the task ball if the controlled ball $\mathbb{O}(\hat{x}_{s(t)}(t), \chi(t))$ fully goes into the task ball $\mathbb{O}(c_{\eta+1}, R_{\eta+1})$, i.e., $\chi(t) + \left\|\hat{x}_{s(t)}(t) - c_{\eta+1}\right\| \leq R_{\eta+1}$, $\eta = 0, 1, \ldots, k_* - 1$. The second method is to find a sequence of $\varphi(t)$ such that $\|x(t) - r(t)\| \leq \varphi(t)$. Then the state stays in the ball $\mathbb{O}(c_{\eta+1}, R_{\eta+1})$, if $\varphi(t) + \|r(t) - c_{\eta+1}\| \leq R_{\eta+1}$. Since we aim to make sure the system state $x(t)$ stays inside the ball for $\mathbb{T}_{\eta+1}$ time instants at least, we propose an event-triggered task-switching control in Algorithm 1.

Algorithm 2 Distributed filter with predicted control

Initial setting: Under the same initial setting as Algorithm 1, set the initial estimate $\hat{x}_i(0)$ and the filter gain G_i satisfying Assumption 2.1, the parameter $\alpha \in (0, \frac{2}{\lambda_{max}(\mathcal{L})})$, the communication step L satisfying (7);
For $t = 1, 2, \ldots$
  Predicted Controller:
  Each sensor i uses t_η from the center,

$$\hat{u}_i(t) = \begin{cases} 0, & \text{if } t \leq T_u \\ K\left(\hat{x}_i(t) - r(t)\right) + u_r(t), & \text{otherwise} \end{cases}$$

  where $r(t)$ and $u_r(t)$ are given in (5).
  Estimation Update: for each sensor i
  $\tilde{x}_i(t+1) = A\hat{x}_i(t) + B\hat{u}_i(t) + G_i(y_i(t) - C_i\hat{x}_i(t))$
  Estimation Consensus for L **steps:** $\bar{x}_{i,0}(t+1) = \tilde{x}_i(t+1)$
   For $l = 1, \ldots, L$
    Sensor i receives $\bar{x}_{j,l-1}(t+1)$ from neighbor j

$$\bar{x}_{i,l}(t+1) = \bar{x}_{i,l-1}(t+1) - \alpha \sum_{j\in\mathcal{N}_i} (\bar{x}_{i,l-1}(t+1) - \bar{x}_{j,l-1}(t+1))$$

   EndFor
  Output step: $\hat{x}_i(t+1) = \bar{x}_{i,L}(t+1)$.
  If $i = s(t)$
   Sensor i transmits its estimate to the control center
  EndIf
EndFor

3.2 *Distributed filter with predicted control*

In this subsection, we propose a two time-scale distributed filter based on a predicted control law in Algorithm 2, which is used by each sensor in the filter block of Fig. 2.

The parameter matrix $G_i \in \mathbb{R}^{n\times M_i}$ stands for the filtering gain. The integer L stands for the communication times of neighboring sensors between two observation updates. The parameter $\alpha \in \mathbb{R}$ is a consensus parameter, which influences the consensus speed of state estimates of sensors. It can be proved that all the estimates over a connected sensor network will reach consensus if L goes to infinity and $\alpha \in (0, \frac{2}{\lambda_{max}(\mathcal{L})})$. In this paper, L is not necessarily large, whose requirement will be studied in the sequel. For sensor $s(t)$, which transmits its estimates to the center controller, its prediction control equals to the true control. Then, its state estimation error is mainly resulted from the steps of observation update and neighboring consensus. For other sensors, e.g., sensor $i, i \neq s(t)$, besides the above two steps as sensor $s(t)$, the estimation error is contributed by inexact control input, which is resulted from the difference between estimates of sensor i and sensor $s(t)$. For control-free systems without tasks, i.e., $u(t) \equiv 0$, the filter can be used to estimate the state $x(t)$ by setting $\hat{u}_i(t) \equiv 0$.

The following lemma, obtained by using (1) under Assumptions 2.1–2.2, is provided to show the trajectory tracking error at the initial control time T_u.

***Lemma** 3.1.* At the start control time T_u, the trajectory tracking error satisfies

$$\|x(T_u) - r_{0,1}(1)\| \leq \left\|A^{T_u}\right\| q_r + \left\|(A^{T_u} - I_n)r_{0,1}(1)\right\| + \sum_{j=0}^{T_u-1} \left\|A^j\right\| q_w =: R(T_u).$$

4. PERFORMANCE ANALYSIS

In this section, we study the performance of Algorithms 1 and 2. We aim to find the conditions such that the trajectory tracking error and the state estimation error are both bounded.

4.1 *Estimation performance of Algorithm 2*

***Lemma** 4.1.* Under Assumption 2.1, $A - GC$ is Schur stable, then the algebraic Riccati equation $(A-GC)^TP(A-GC) + I_n = P$ has a positive definite matrix solution $P = \sum_{i=0}^{\infty}((A-GC)^i)^T(A-GC)^i$.

The boundedness of the state estimation error is studied in the following theorem.

Theorem 4.1. Consider the system (1) satisfying Assumptions 2.1 and 2.2. For Algorithm 2, if

$$L \geq \lceil L_0 \rceil, \tag{7}$$

in which L_0 is in (6), and $\rho > 0, \tau > 0$ and $\gamma > 0$ are subject to

$$\begin{aligned} &(1+\rho)(1+\tau) \leq 1 + \frac{1}{3\left(\lambda_{\max}\left(P\right) - 1\right)}, \\ &\gamma \geq \gamma_0, \end{aligned} \tag{8}$$

then the following results hold:

(i) The estimation error $e_i(t)$ is bounded, i.e.,
$$\|e_i(t)\| \leq \chi(t), \forall i \in \mathcal{V}, \forall t \geq 0; \tag{9}$$
(ii) The estimation error is asymptotically upper bounded, i.e.,
$$\limsup_{t\to\infty} \|e_i(t)\| \leq \sqrt{\frac{6d_0\lambda_{\max}(P)}{\min\{N,\gamma^2\}\lambda_{\min}(P)}}; \tag{10}$$
(iii) Furthermore, if the system is noise-free, then
$$\limsup_{t\to\infty} \|e_i(t)\| = 0, \forall j \in \mathcal{V},$$

where P is in Lemma 4.1, γ_0 , $\bar{M}_{1,1}$, $\bar{M}_{2,2}(t)$, $\chi(t)$, d_0, and the matrices therein are all given in (6).

4.2 *Control performance of Algorithm 1*

Recall that t_η is the task-switching time from task η to $\eta+1$, then we study the boundedness of the trajectory tracking error in the following theorem.

Theorem 4.2. Under the same conditions as in Theorem 4.1. Then the Riccati equation $(A+BK)^TP_*(A+BK)+I_n = P_*$ has a solution $P_* \succ 0$, such that for $0 = 1, \ldots, k_* - 1$, the following results hold

(i) The trajectory tracking error is upper bounded, i.e.,
$$\|x(t) - r(t)\| \leq \varphi(t), \forall t \in [t_\eta, t_{\eta+1}); \tag{11}$$
(ii) If $\{r(t)\}$ is infinite and the control center does not generate new switching signal, then the trajectory tracking error is asymptotically upper bounded, i.e.,
$$\limsup_{t\to\infty} \|x(t) - r(t)\| \leq \frac{\beta\left(\|BK\|\sqrt{\frac{6d_0\lambda_{\max}(P)}{\min\{N,\gamma^2\}\lambda_{\min}(P)}} + q_w\right)^2}{\lambda_{\min}(P_*)(1-\lambda)}; \tag{12}$$
(iii) Under the same conditions as in (ii), if the system is noise-free, then the trajectory tracking error tends to zero, i.e.,
$$\limsup_{t\to\infty} \|x(t) - r(t)\| = 0, \tag{13}$$

$$
\begin{aligned}
L_0 &= \max\left\{\frac{\ln\left(\gamma\kappa\sqrt{3(1+\rho)(1+\tau)}\right)}{\ln\left(\lambda_c^{-1}\right)}, \frac{\ln\left((\max\{\|A+BK\|,\|A\|\}+\kappa)\sqrt{3(1+\rho)(1+\frac{1}{\tau})}\right)}{\ln\left(\lambda_c^{-1}\right)}\right\},\\
\kappa &= \left\|(I_{Nn}-P_{Nn})\bar{G}\bar{C}\right\|, \qquad \lambda_c = \|I_{Nn}-\alpha(\mathcal{L}\otimes I_n)-P_{Nn}\| \in (0,1), \quad P_{Nn} = \frac{1}{N}(\mathbf{1}_N\otimes I_n)(\mathbf{1}_N\otimes I_n)^T,\\
\gamma_0 &= \sqrt{3(1+\rho)(1+\frac{1}{\tau}) \sup_{s(t)\in\mathcal{V},t\geq 0} \lambda_{max}\left(\bar{M}^T(s(t),t)\left(I_N\otimes P\right)\bar{M}(s(t),t)\right)},\\
\bar{M}(s(t),t) &= \mathbf{1}_N\otimes\left(BK\mathcal{I}_{t\geq T_u}(m_{s(t)}\otimes I_n)+\frac{1}{N}(\mathbf{1}_N^T\otimes I_n)\bar{G}\bar{C}\right), \qquad (6)\\
\chi(t) &= \sqrt{\frac{2}{\min\{N,\gamma^2\}\lambda_{\min}(P)}\left(\lambda_{\max}(P)Nq_x^2(1+4\gamma^2)\varpi^t+d_0\frac{1-\varpi^t}{1-\varpi}\right)}, \qquad \varpi = 1-\frac{1}{3\lambda_{\max}(P)} \in \left[\frac{2}{3},1\right)\\
d_0 &= (1+\frac{1}{\rho})\lambda_{\max}(P)N\left(\left(q_w+q_v\sum_{i=1}^{N}\|G_i\|\right)^2+q_v^2\gamma^2\lambda_c^2\left\|(I_{Nn}-P_{Nn})\bar{G}\right\|^2\right)\\
\varphi(t) &= \sqrt{\frac{\lambda_{\max}(P_*)\left(R_\eta\mathcal{I}_{\eta\geq 1}+R(T_u)\mathcal{I}_{\eta=0}\right)^2}{\lambda_{\min}(P_*)}\lambda^{t-t_\eta}+\frac{\beta}{\lambda_{\min}(P_*)}\sum_{l=0}^{t-t_\eta-1}\lambda^{t-t_\eta-1-l}\left(\|BK\|\chi(t_\eta+l)+q_w\right)^2}.
\end{aligned}
$$

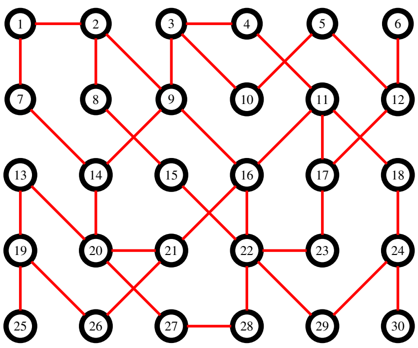


Fig. 3. A sensor network with 30 sensors

where $\lambda = 1-\frac{1}{2\lambda_{\max}(P_*)} \in (0,1)$, $\beta = \|P_*\| + 2\|P_*(A+BK)\|^2$, $\varphi(t)$ and $\chi(t)$ are in (6).

5. NUMERICAL SIMULATION

In this section, we study some numerical simulations to test the performance of the proposed framework and verify the theoretical results.

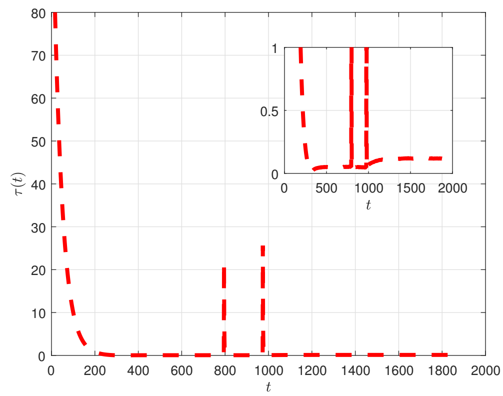
Consider an undirected and connected sensor network with 30 sensors, which is illustrated in Fig. 3. The topology of the sensor network provides: $\lambda_2(\mathcal{L}) = 0.17$ and $\lambda_{max}(\mathcal{L}) = 6.66$, where $\mathcal{L}$ is the Laplacian matrix of the network. For the system (1), we assume that $B = I_2$, A and C_i, $i = 1,\ldots,30$, are chosen in the following manner

$$
A = \begin{pmatrix} 1.01 & 0.05 \\ 0 & 1.01 \end{pmatrix}, C_i = \begin{cases} (1,1), & \text{if } 1\leq i\leq 10\\ (1,0), & \text{if } 11\leq i\leq 20\\ (0,1), & \text{if } 21\leq i\leq 30. \end{cases}
$$

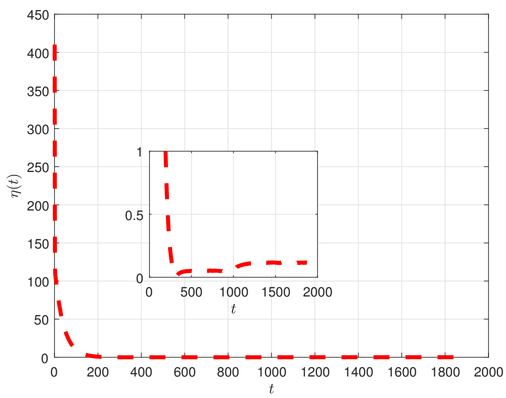
Consider the scenario in Fig. 1, where three task balls $\{\mathbb{O}(c_i,R_i)\}_{i=1}^3$ and an initial error ball $\mathbb{O}(c_0,R_0)$ exist, which are connected by three reference trajectories (i.e., $r(t)$). The parameters of the four balls are given:

$c_0 = (0,0)^T, c_1 = (450,-50)^T, c_2 = (550,500)^T,$

$c_3 = (50,400)^T, R_0 = 200, R_1 = 130, R_2 = 150, R_3 = 120.$



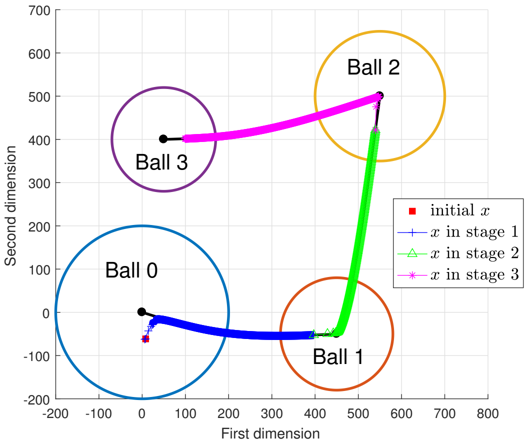
(a) Trajectory tracking error by Algorithm 1.



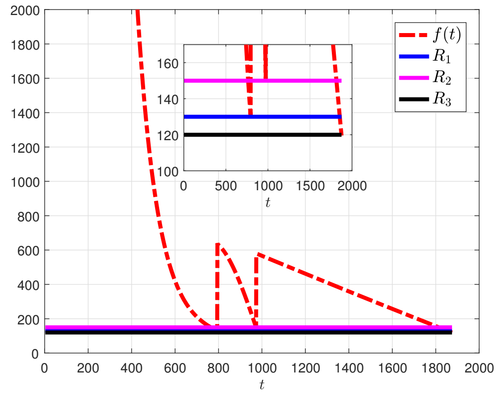
(b) State estimation error by Algorithm 2.

Fig. 4. Performance of Algorithms 1 and 2.

The initial state $x(0)$ is generated by the manner: $x(0) = (\bar{R}_0\cos(\bar{\theta}_0), \bar{R}_0\sin(\bar{\theta}_0))$, where $\bar{R}_0$ and $\bar{\theta}_0$ are uniformly distributed in the interval $[0,R_0]$ and $[0,2\pi]$, respectively. It ensures $x(0) \in \mathbb{O}(c_0,R_0)$. The tracking sequence for the task balls is $\{1,2,3\}$, by following the reference trajectories from



(a) Trajectory tracking error by Algorithm 1.



(b) Event-triggered bound and switching times.

Fig. 5. Trajectory tracking and task-switching times. The dynamics of $f(t)$ in Algorithm 1 and $g(t) := \varphi(t) + \|r(t) - c_\eta\|$ are given in (b). The sequence of $f(t)$ is obtained online, while $g(t)$ can be calculated offline to estimate the task switching times.

balls 0 to 1, 1 to 2, and 2 to 3. The three processes are called the first, second, and third stages, respectively, whose lengths are as follows: 900, 200, 1000. The label of the sensor $s(t)$, which sends its estimates to the centralized control center, is switched: $s(t) = 1$ if t belongs to the first stage, $s(t) = 15$ if t belongs to the second stage, and $s(t) = 30$ if t belongs to the third stage. We conduct a Monto Carlo experiment with 100 runs. Define the estimation error $\eta(t)$ and the trajectory tracking $\tau(t)$ as follows $\eta(t) = \frac{1}{100}\sum_{j=1}^{100}\left\|\hat{x}^j_{s(t)}(t) - x^j(t)\right\|$, and $\tau(t) = \frac{1}{100}\sum_{j=1}^{100}\left\|x^j(t) - r(t)\right\|$, where $x^j(t)$ and $\hat{x}^j_{s(t)}(t)$ are the system state and the state estimate by sensor $s(t)$ (i.e., the estimate transmitted to the control center) at time t in the j-th run, respectively.

The initial estimates of all sensors are zeros. Suppose that $w(t)$ and $v_i(t)$ follow the uniform distribution with $q_w = q_v = 0.001$. Let $K = \mathrm{diag}([-0.6, -0.7])$ and $G = [0.34, 0.54]^T$ which satisfy the requirements in Assumption 2.1. Suppose the desired least remaining time in each task ball is $\mathbb{T}_i = 1$, and the start control time is $T_u = 0$. By using Algorithm 1, and Algorithm 2 with $L = 1$ and $\alpha = \frac{2}{\lambda_2(\mathcal{L}) + \lambda_{max}(\mathcal{L})} = 0.29$, the tracking error, the state estimation error by sensor $s(t)$, and the dynamics of the event-triggered bound $f(t)$ in Algorithm 1, are obtained in Fig. 4 and Fig. 5. It shows the estimation error tends to a small neighborhood of zero. Also, the trajectory tracking error tends to a small neighborhood of zero, except for two short periods, which occurs since the task switches at the times 794 and 973. The dynamics of $f(t)$ show two fluctuation periods as well, because the reference trajectory is switched from 1 to 2, and 2 to 3 at the times 794 and 973 at which the event-triggered task-switching condition is met. The state dynamics are given in Fig. 5, through which we see the state follows the reference trajectories closely except in three transient periods.

6. CONCLUSIONS

This paper studied the problem that how to control an agent over a connected sensor network such that the agent state is able to reach certain task sets in order by tracking multiple reference trajectories. A framework of an event-triggered control framework based on distributed estimation was proposed under constrained communication between a sensor network and the agent. The state estimation error and the trajectory tracking error were both asymptotically upper bounded.

REFERENCES

Aguiar, A.P. and Hespanha, J.P. (2007). Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty. *IEEE Transactions on Automatic Control*, 52(8), 1362–1379.

Antonelli, G., Arrichiello, F., Caccavale, F., and Marino, A. (2014). Decentralized time-varying formation control for multi-robot systems. *The International Journal of Robotics Research*, 33(7), 1029–1043.

Franchi, A., Petitti, A., and Rizzo, A. (2015). Decentralized parameter estimation and observation for cooperative mobile manipulation of an unknown load using noisy measurements. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 5517–5522. IEEE.

Freundlich, C., Lee, S., and Zavlanos, M.M. (2017). Distributed active state estimation with user-specified accuracy. *IEEE Transactions on Automatic Control*, 63(2), 418–433.

Li, S., Kong, R., and Guo, Y. (2014). Cooperative distributed source seeking by multiple robots: Algorithms and experiments. *IEEE/ASME Transactions on Mechatronics*, 19(6), 1810–1820.

Marino, A. (2017). Distributed adaptive control of networked cooperative mobile manipulators. *IEEE Transactions on Control Systems Technology*, 26(5), 1646–1660.

Miao, Z., Liu, Y.H., Wang, Y., Yi, G., and Fierro, R. (2018). Distributed estimation and control for leader-following formations of nonholonomic mobile robots. *IEEE Transactions on Automation Science and Engineering*, 15(4), 1946–1954.

Roumeliotis, S.I. and Bekey, G.A. (2002). Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5), 781–795.

Wang, Z. and Gu, D. (2011). Cooperative target tracking control of multiple robots. *IEEE Transactions on Industrial Electronics*, 59(8), 3232–3240.

Yang, P., Freeman, R.A., and Lynch, K.M. (2008). Multi-agent coordination by decentralized estimation and control. *IEEE Transactions on Automatic Control*, 53(11), 2480–2496.