2023 European Control Conference (ECC)
June 13-16, 2023. Bucharest, Romania

Navigating A Mobile Robot Using Switching Distributed Sensor Networks

Xingkang He, Ehsan Hashemi, and Karl H. Johansson

***Abstract*— This paper proposes a method to navigate a mobile robot by estimating its state over a number of distributed sensor networks (DSNs) such that it can successively accomplish a sequence of tasks, i.e., its state enters each targeted set and stays inside no less than the desired time, under a resource-aware, time-efficient, and computation- and communication-constrained setting. We propose a new robot state estimation and navigation architecture, which integrates an event-triggered task-switching feedback controller for the robot and a two-time-scale distributed state estimator for each sensor. With the controller, the robot is able to accomplish a task by following a reference trajectory and switch to the next task when an event-triggered condition is fulfilled. With the estimator, each active sensor is able to estimate the robot state. We provide conditions to ensure that the state estimation error and the trajectory tracking deviation are upper bounded by two time-varying sequences, respectively. Furthermore, we find a sufficient condition for accomplishing a task and provide an upper bound of running time for the task. Numerical simulations of an indoor robot's localization and navigation are provided to validate the proposed architecture.**

I. Introduction

Autonomous mobile robots with augmented state estimation and navigation systems over sensor networks are revolutionizing accurate navigation and controls for indoor and outdoor applications, such as service robots in dynamic environments, automated storage/retrieval with mobile robots in a warehouse [1]. In addition, in cooperative intelligent transportation systems, reliable navigation through vehicle-to-infrastructure and over sensor networks, plays a vital role in reliable decision-making, safe motion planning, and controls for automated driving in urban settings [2].

Mobile robot navigation has been intensively studied through centralized frameworks [3]. In contrast, state estimation over distributed sensor networks (DSNs) shows advantages in structure robustness and parallel data processing, and thus draws more and more attention in recent years [4], [5]. However, few results are given on the design of distributed estimators based feedback controllers. Since timely control signal synchronization between control center and DSNs induces a large amount of communications, new architectures able to alleviate communications between the center and DSNs are expected. Moreover, the number of sensors in a DSN can lead to a tradeoff between resource consumption and estimation performance. On one hand, running a large-size DSN with many redundant sensors may lead to a wast of resources, especially when the desired estimation performance can be ensured with a small-size DSN. On the other hand, a too small-size DSN may provide insufficient information for the desired estimation performance. Thus, in different scenarios, how to timely choose proper DSNs balancing resource consumption and estimation performance needs investigation. Besides, how to coordinate the switching of DSNs is of interest.

This paper studies how to navigate a mobile robot by estimating its state over a number of DSNs such that the robot is able to successively accomplish a sequence of tasks, i.e., its state enters each targeted set and stays inside no less than the desired time, in a resource-aware, time-efficient, and computation- and communication-constrained setting.

The remainder of the paper is organized as follows. Section II is on problem formulation. Section III provides the design of the architecture, the controller, and the estimator. Section IV analyzes the main properties of the controller and the estimator. Numerical simulations are given in Section V. Section VI concludes this paper. Due to space constraint, the proofs and some results are given in the extended version [6].

Notations: $\|A\|$ is the Euclidean norm. $\mathbf{1}_N$ stands for the N-dimensional vector with all elements being one. For integers m and n with $m < n$, let $[m, n] = \{m, m+1, \ldots, n\}$. $|S|$ denotes the cardinality of set S.

II. Problem Formulation

An example is provided in Fig. 1 to illustrate the problem studied in this paper.

A. Model of the controlled robot

Consider a mobile robot with the following system dynamics

$$x(t+1) = Ax(t) + Bu(t) + w(t), \quad t \in \mathbb{N}^+ \qquad (1)$$

where $x(t) \in \mathbb{R}^n$ is the unknown robot state, $u(t) \in \mathbb{R}^p$ the control input, $w(t) \in \mathbb{R}^n$ the bounded process uncertainty, i.e., there is a scalar $q_w \geq 0$, such that $\|w(t)\| \leq q_w$, which can be used to model linearization error, unknown disturbance and uncertain dynamics. $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times p}$ are known system matrices. For more information of linear modeling of mobile robots as in (1), readers can refer to [7].

Assumption 1: The robot system is stabilizable, i.e., there is a matrix $K \in \mathbb{R}^{p\times n}$ such that $A + BK$ is Schur stable.

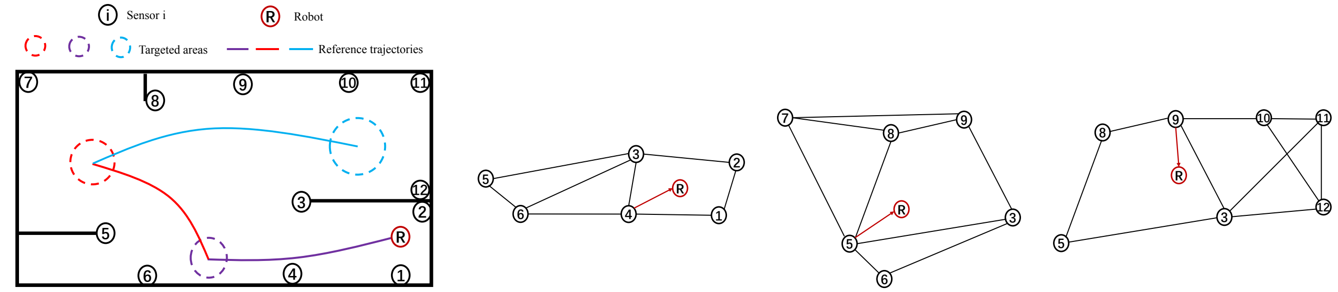
The work is supported by the Knut & Alice Wallenberg Foundation and the Swedish Research Council.

X. He is with Ericsson AB, Sweden. The majority of this paper was done when he worked at KTH (e-mail: xingkang0715@gmail.com).

K. H. Johansson is with Division of Decision and Control Systems, School of Electrical Engineering and Computer Science. KTH Royal Institute of Technology, Sweden (e-mail: kallej@kth.se).

E. Hashemi is with Department of Mechanical Engineering, University of Alberta, Canada (e-mail: ehashemi@ualberta.ca).

978-3-907144-08-4 ©2023 EUCA



(a) A schematic of robot localization and navigation with twelve powered sensors (b) Active graph for task 1 (c) Active graph for task 2 (d) Active graph for task 3

Fig. 1. In a building with twelve powered sensors, a mobile robot aims at successively accomplishing three tasks, i.e., moving into three targeted areas in order and staying inside at least for the desired time by following three reference trajectories, respectively, as shown in (a). The robot is able to activate a subset of the twelve sensors for the localization in each task. The active sensors in each task form a DSN to sense and estimate the robot state collaboratively in order to provide the localization information for the robot, as shown in (b)–(d) where the robot only receives messages from a single active sensor.

We aim to design $u(t)$ such that the robot is able to accomplish a sequence of tasks defined as follows.

***Definition** 1:* The robot has accomplished task $\eta \in \mathbb{N}^+$ at time $t \in \mathbb{N}^+$, if there is a time t_η with $t \geq t_\eta \geq \mathbb{T}_\eta$, such that $x(\bar{t}) \in \mathbb{O}(c_\eta, R_\eta)$, for $\forall \bar{t} \in [t_\eta - \mathbb{T}_\eta, t_\eta]$ where $\mathbb{O}(c_\eta, R_\eta) = \{x \in \mathbb{R}^n | \, \|D_\eta x - c_\eta\| \leq R_\eta\}$ is the η-th targeted set, $\mathbb{T}_\eta \in \mathbb{N}^+$ the dwell time, $c_\eta \in \mathbb{R}^{d_\eta}$ and $R_\eta \in \mathbb{R}^+$ the center and radius of the targeted set, $D_\eta \in \mathbb{R}^{d_\eta \times n}$ the constraint matrix, and d_η a positive integer less than n.

Remark 1: Because of process uncertainties, it makes sense to define a task with a targeted set but not a point. If dwell time $\mathbb{T}_\eta = 1$, the problem reduces to the controllability problem under uncertainties. For different task η, parameters $\{\mathbb{T}_\eta, D_\eta, c_\eta, R_\eta\}$ may be totally differently. The defined task can reduce to different tasks in different problems. For example, if $D_\eta = I$ and $c_\eta = 0$, the task reduces to a stabilization task. For the second-order system with $A = \left(\begin{smallmatrix} 1 & T \\ 0 & 1 \end{smallmatrix}\right)$ and $T > 0$, if $D_\eta = (1, 0)$, the task becomes how to control the robot's position into a certain area. A targeted set in Definition 1 essentially refers to a multi-dimensional ball. The proposed methods in this paper can still be used for other types of targeted sets by partitioning and approximating them via a number of such balls.

Suppose there are $d \in \mathbb{N}^+$ task(s) with task order $\eta = 1, 2, \ldots, d$. For convenience, task $\eta = 0$ stands for the initialization. If task $\eta = 0, \ldots, d-1$ is accomplished, we need a reference trajectory to navigate the robot from targeted set $\mathbb{O}(c_\eta, R_\eta)$ to targeted set $\mathbb{O}(c_{\eta+1}, R_{\eta+1})$. For $\eta = 0, 1, \ldots, d-1$, suppose there is a reference trajectory, consisting of reference states $\{r_{\eta,\eta+1}(l)\}_{l=1}^{T_{\eta,\eta+1}}$ and reference inputs $\{u_{\eta,\eta+1}(l)\}_{l=1}^{T_{\eta,\eta+1}}$, subject to

$$\begin{aligned} D_{\eta+1} r_{\eta,\eta+1}(1) = c_\eta, \quad D_{\eta+1} r_{\eta,\eta+1}(T_{\eta,\eta+1}) = c_{\eta+1}, \\ r_{\eta,\eta+1}(l+1) = A r_{\eta,\eta+1}(l) + B u_{\eta,\eta+1}(l), \end{aligned} \tag{2}$$

where $T_{\eta,\eta+1} \in \mathbb{N}^+$ is the length of the $(\eta+1)$-th reference trajectory, and $r_{0,1}(1)$ is the initial reference state to be determined in Assumption 4.

Remark 2: The reference trajectories can be generated with interpolation methods (e.g., cubic splin interpolation as used in the simulation) or trajectory optimization, e.g., signal temporal logic in formal methods [8] and direct collocation methods [9]. Due to system uncertainties, it is not suggestible to replace control input $u(t)$ by the reference inputs (as shown in the simulation).

B. Model of sensor networks

There are $N \geq 2$ (smart and powered) sensors which can be activated to measure and estimate the robot's state. Due to physical limitation or resource constraints, only a subset of sensors are active at each time for sensing and estimation, and the rest of sensors remain inactive. The measurement equation of sensor i at time t is:

$$y_i(t) = \delta_i(t) \left(C_i x(t) + v_i(t)\right), i = 1, \ldots, N, \tag{3}$$

where $y_i(t) \in \mathbb{R}^{M_i}$ is the measurement vector, $v_i(t) \in \mathbb{R}^{M_i}$ the measurement uncertainty, $C_i \in \mathbb{R}^{M_i \times n}$ the measurement matrix, and $\delta_i(t) \in \{0, 1\}$ is an indicator of sensor's state, where $\delta_i(t) = 1$ if sensor i is active at time $t \in \mathbb{N}^+$, otherwise $\delta_i(t) = 0$. Assume the robot can determine the values of $\{\delta_i(t)\}$ via communicating with sensors. Suppose $v_i(t)$ is bounded, i.e., there is a scalar $q_v \geq 0$, such that $\|v_i(t)\| \leq q_v$. Such measurement uncertainty can be used to model sensor bias, unknown disturbance, and bounded noise [10]. Assume that each sensor is able to store a set of parameters and the reference trajectory data $\{r_{\eta,\eta+1}(l)\}_{l=1}^{T_{\eta,\eta+1}}$, $\{u_{\eta,\eta+1}(l)\}_{l=1}^{T_{\eta,\eta+1}}$ with $\eta = 0, 1, \ldots, d-1$ in advance.

Since an individual sensor has limited observability (i.e., (A, C_i) is unobservable or undetectable), the sensors are expected to estimate the robot state collaboratively. Suppose the N sensors form Q_s DSNs. We model the communication of the k-th network with $2 \leq N(k) \leq N$ nodes through a fixed undirected graph $\mathcal{G}(k) = (\mathcal{V}(k), \mathcal{E}(k), \mathcal{A}(k))$, where $\mathcal{V}(k) \subset \{1, 2, \ldots, N\}$ denotes the set of nodes, $\mathcal{E}_k \subseteq \mathcal{V}(k) \times \mathcal{V}(k)$ the set of edges, and $\mathcal{A}(k)$ the 0–1 adjacency matrix. $\mathcal{L}(k)$ denotes the Laplacian matrix. The following assumptions are needed in this paper.

***Assumption** 2:* To accomplish d tasks in order, the robot is able to activate a sequence of connected graphs $\{\mathcal{G}_\eta\}_{\eta=1}^{d}$ with $\mathcal{G}_\eta = \{\mathcal{V}_\eta, \mathcal{E}_\eta, \mathcal{A}_\eta\} \in \{\mathcal{G}(k)\}_{k=1}^{Q_s}$, such that the system is collectively detectable over $\mathcal{G}_\eta$ for each task η.

The collective detectability in Assumption 2 means that for each η, there is a matrix G_η such that $A - G_\eta \tilde{C}_\eta$ is Schur stable, where $\tilde{C}_\eta$ is obtained by stacking all the measurement matrices of sensors in the set $\mathcal{V}_\eta$. The collective detectability is satisfied if there is one sensor i such that (A, C_i) is detectable, but not vice versa. The detectability condition reduces to the mildest one in distributed estimation when network $\mathcal{G}_\eta$ includes all sensors. Even if the system is not collectively detectable for some task η, one can partition this task into several tasks with length-reduced reference trajectories, then Assumption 2 is satisfied as long as in each new task the collective detectability is ensured.

Assumption 3: The robot has the full knowledge of the system and sensor networks $\{\mathcal{G}_\eta\}_{\eta=1}^d$ satisfying Assumption 2.

Assumption 3 enables the robot to have sufficient information such that it can make decisions of when to switch to the next task.

Assumption 4: At the initial time, it holds that $\|x(1) - \hat{x}_i(1)\| \le q_x, \|x(1) - r_{0,1}(1)\| \le q_r$, where q_x and q_r are non-negative scalars, $\hat{x}_i(1)$ is the estimate of $x(1)$ by sensor i, $i = 1, 2 \ldots, N$, and $r_{0,1}(1)$ is the initial reference state from (2).

C. *Problem of interest*

We focus on solving the following subproblems: 1) How to design an integrated estimation and control architecture such that the robot is able to successively accomplish d tasks by switching the active DSN? (Section III); 2) How to evaluate the estimation error and trajectory tracking deviation in each task? 3) What conditions can ensure that each task is successfully accomplished?

III. Task-Switching Architecture

In this section, we design an integrated task-switching architecture, consisting of an event-triggered feedback controller for the robot and a distributed estimator for each active sensor.

A. *Integrated estimation and control architecture*

Let $\mathcal{G}_\eta = (\mathcal{V}_\eta, \mathcal{E}_\eta, \mathcal{A}_\eta)$ satisfying Assumption 2 be the active sensor network for task $\eta \ge 1$. Then the set of active sensors for task η or $\eta + 1$ is denoted as

$$\bar{\mathcal{V}}(\eta) = \mathcal{V}_{\eta-1} \bigcup \mathcal{V}_\eta. \quad (4)$$

Suppose $\Delta_{\eta-1}(t) \ge 0$ is the condition such that the task is switched from $\eta - 1$ to η at time t. Then at this time instant, the robot broadcasts to each sensor in the set $\bar{\mathcal{V}}(\eta)$ with the following message

$$\mathcal{M}_\eta = \{\eta, \hat{x}_{s_{\eta-1}}(t), s_\eta\}, \quad (5)$$

where η is the next task index, and $\hat{x}_{s_{\eta-1}}(t)$ is the latest estimate of the robot received from sensor $s_{\eta-1}$ in task $\eta - 1$, and s_η is the label of the sensor transmitting estimates to the robot in task η.

The whole architecture is depicted in Fig. 2. Communication in this architecture exists in three aspects: (i). The

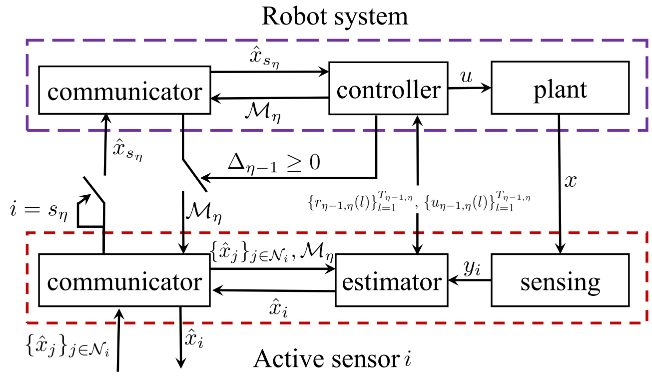


Fig. 2. An integrated estimation and control architecture with event-triggered task switching. When $\Delta_{\eta-1} \ge 0$, the robot broadcasts $\mathcal{M}_\eta$ to sensor $i \in \bar{\mathcal{V}}(\eta)$. If sensor i knows from $\mathcal{M}_\eta$ that it should be active in this task, to update its estimate $\hat{x}_i$, it will use measurement y_i, $\mathcal{M}_\eta$, the reference trajectory data $\{r_{\eta-1,\eta}(l)\}_{l=1}^{T_{\eta-1,\eta}}$ and $\{u_{\eta-1,\eta}(l)\}_{l=1}^{T_{\eta-1,\eta}}$, and the estimates from neighbor active sensors. If sensor i is inactive in this task, it will shut down the estimator and sensing components until it is activated again. Sensor s_η communicates its estimate $\hat{x}_{s_\eta}$ with the robot for the controller design.

robot communicates message $\mathcal{M}_\eta$ to a subset of sensors ($\bar{\mathcal{V}}(\eta)$) only at the moment when the task is switched. (ii). The robot receives its state estimates from one sensor. The sensor label s_η could be different in different task η. For example, a mobile robot can choose a near sensor to obtain its state estimate. (iii). The active sensors can communicate with each other in a distributed manner over the activated sensor network. Since the existing centralized architectures often require that the exact control signal $u(t)$ is shared with sensors all the time and that the robot communicates with all sensors for obtaining their measurements, this architecture is more suitable when the system is under constrained resources or communication.

B. *Task-switching feedback control*

For task $\eta \ge 1$, according to the architecture in Fig. 2, we design the following feedback control input $u(t)$ with a fixed control gain $K \in \mathbb{R}^{p \times n}$:

$$\begin{aligned} u(t) &= K\left(\hat{x}_{s_\eta}(t) - r(t)\right) + u_r(t), \\ r(t) &= r_{\eta-1,\eta}(t - t_{\eta-1} + 1), \\ u_r(t) &= u_{\eta-1,\eta}(t - t_{\eta-1} + 1), \end{aligned} \quad (6)$$

where $t_{\eta-1}$ is the ending time for task $\eta - 1$ (also the initial time for task η), i.e., the time when $\Delta_{\eta-1}(t) \ge 0$.

Remark 3: Since $u(t)$ in (6) does not involve complex computations, it is suitable for the robot with limited computational resources. Compared with $u_r(t)$ as the control input signal, feedback control signal $u(t)$ in (6) has advantages against system uncertainties.

Condition $\Delta_{\eta-1}(t) \ge 0$ is supposed to ensure that task $\eta - 1$ is accomplished for sure. The following lemma provides a sufficient condition such that the robot is in a targeted set at a single time.

Algorithm 1 Event-triggered task-switching controller

Initial setting: Control gain K, targeted sets $\{\mathbb{O}(c_\eta, R_\eta)\}_{\eta=1}^d$, reference states $\{r_{\eta-1,\eta}(l)\}_{l=1}^{T_{\eta-1,\eta}}$ and inputs $\{u_{\eta-1,\eta}(l)\}_{l=1}^{T_{\eta-1,\eta}}$, $\eta = 1,\ldots,d$, desired dwell time in targeted sets $\{\mathbb{T}_\eta\}$, cumulative remaining time set $\{\mathcal{T}_\eta\}$ initialized by empty sets, task accomplishment indicators $\mathcal{J}_\eta = 0$ for each η, initial task label $\eta = 1$;

for $t = 1, 2, \ldots$ **do**
- **// Communication with sensor s_η:**
- Robot obtains an estimate $\hat{x}_{s_\eta}(t)$ from sensor s_η;
- **// Event-triggered condition for task switching:**
- **if** $f(t) \le R_\eta$ **then**
  - $\mathcal{T}_\eta = \mathcal{T}_\eta \bigcup \{t\}$
- **end if**
- **if** $|\mathcal{T}_\eta| = \mathbb{T}_\eta$ **then**
  - $\mathcal{J}_\eta = 1$ // task η is accomplished
  - $t_\eta = t$ // task switching time instant
- **end if**
- **// Communication with some sensors when task η is accomplished**
- **if** $\mathcal{J}_\eta = 1$ **then**
  - Broadcast message $\mathcal{M}_{\eta+1}$ in (5) to sensors in the set $\bar{\mathcal{V}}(\eta+1)$ (4);
  - $\eta = \eta + 1$;
- **end if**
- **// Feedback control input:**
- Design $u(t)$ as in (6);

end for

Algorithm 2 Task-switching distributed estimator

Initial setting: Control gain K, reference states $\{r_{\eta-1,\eta}(l)\}_{l=1}^{T_{\eta-1,\eta}}$ and inputs $\{u_{\eta-1,\eta}(l)\}_{l=1}^{T_{\eta-1,\eta}}$, $\eta = 1,\ldots,d$, parameter set $\{G_{i,\eta}, L_\eta, \alpha_\eta\}_{\eta=1}^d$, and initial estimate $\hat{x}_i(1)$;

for $t = 1, 2, \ldots$ **do**
- **if** Sensor i receives $\mathcal{M}_\eta$ and $i \notin \mathcal{V}_\eta$ **then**
  - Sensor i becomes inactive in sensing and estimation;
- **else if** Sensor i receives $\mathcal{M}_\eta$ and $i \in \mathcal{V}_\eta$ **then**
  - Based on $\mathcal{M}_\eta$ in (5), update η, s_η, and let $\hat{x}_i(t) = \hat{x}_{s_{\eta-1}}(t)$.
- **else**
  - **// Predicted Controller:**
  - $\hat{u}_i(t) = K(\hat{x}_i(t) - r(t)) + u_r(t)$ where $r(t)$ and $u_r(t)$ are given in (6).
  - **// Estimator Update:**
  - $\tilde{x}_i(t) = A\hat{x}_i(t) + B\hat{u}_i(t) + G_{i,\eta}(y_i(t) - C_i\hat{x}_i(t))$, where $G_{i,\eta}$ is the corresponding gain of sensor i when $\mathcal{G}_\eta$ is active.
  - **// Neighbor sensors communicate for L_η times:**
  - **for** $l = 1,\ldots,L_\eta$ **do**
    - Sensor i receives $\bar{x}_{j,l-1}(t)$ from neighbor sensor j, and runs $\bar{x}_{i,l}(t) = \bar{x}_{i,l-1}(t) - \alpha_\eta \sum_{j\in\mathcal{N}_{i,\eta}} (\bar{x}_{i,l-1}(t) - \bar{x}_{j,l-1}(t))$, where $\bar{x}_{i,0}(t) = \tilde{x}_i(t)$, $\hat{x}_i(t+1) = \bar{x}_{i,L_\eta}(t)$, and $\mathcal{N}_{i,\eta}$ is the neighbor set of sensor i in graph $\mathcal{G}_\eta$;
  - **end for**
  - **// Communication with the robot**
  - **if** $i = s_\eta$ **then**
    - Sensor i transmits its estimate $\hat{x}_{s_\eta}(t+1)$ to the robot.
  - **end if**
- **end if**

end for

Lemma 1: Suppose there are two sequences $\{h_e(t)\}$ and $\{h_c(t)\}$, such that

$$\left\|D_\eta\left(x(t) - \hat{x}_{s_\eta}(t)\right)\right\| \le h_e(t), \quad \|D_\eta(x(t) - r(t))\| \le h_c(t).$$

Then robot state $x(t)$ is in the targeted set $\mathbb{O}(c_\eta, R_\eta)$, if

$$f(t) := \min\left\{h_e(t) + g_e(t), h_c(t) + g_c(t)\right\} \le R_\eta, \qquad (7)$$

where $g_e(t) = \left\|D_\eta \hat{x}_{s_\eta}(t) - c_\eta\right\|$ and $g_c(t) = \|D_\eta r(t) - c_\eta\|$.

Since $\{g_e(t)\}$ and $\{g_c(t)\}$ can be directly computed by the robot, condition (7) can be used in the design of $\Delta_{\eta-1}(t) \ge 0$ provided with two sequences $\{h_e(t)\}$ and $\{h_c(t)\}$ satisfying the requirement in Lemma 1. In Section IV, we provide a design of $\{h_e(t)\}$ and $\{h_c(t)\}$. Based on the proposed architecture in Fig. 2, control input (6), and Lemma 1, we propose an event-triggered task-switching controller (Algorithm 1) such that the robot is able to accomplish the given tasks in sequence.

C. Task-switching distributed estimator

Since Algorithm 1 depends on the state estimate $\hat{x}_{s_\eta}(t)$ from sensor s_η and each active sensor may not have sufficient measurement data for effectively inferring the system state $x(t)$, we aim to design a distributed state estimator over the active DSN in each task η. In Algorithm 2, we propose a two-time-scale distributed estimator for each sensor $i \in \bar{\mathcal{V}}(\eta)$.

Remark 4: Based on the task information (i.e., η in $\mathcal{M}_\eta$) from the robot, each active sensor is able to choose the corresponding reference input $u_r(t)$ and parameters $\{G_{i,\eta}, L_\eta, \alpha_\eta\}$. Integer L_η represents the communication times of each sensor with its neighboring active sensors between two measurement updates. In this paper, L_η is not necessarily very large. For each task, an explicit requirement of L_η is given in Theorem 1. Such a two-time-scale distributed estimation can be supported with the advancement of communication technologies like 5G. In the literature, a number of two-time-scale distributed algorithms have been studied (e.g., [11]).

Remark 5: Since sensor s_η shares its estimate with the robot, its estimation error is due to system uncertainties. For the rest of sensors, their estimation errors are also affected by inexact control inputs. The errors induced from the inexact control inputs will be reduced if a large L_η is designed.

Lemma 2: Under Assumptions 1–3, it is feasible to choose parameters $\{G_{i,\eta}, L_\eta, \alpha_\eta\}_{\eta=1}^d$ for Algorithms 1–2 such that:

- Matrix $A - G_\eta \tilde{C}_\eta$ is Schur stable, where $\tilde{C}_\eta$ and G_η are obtained by stacking measurement matrices $\{C_i\}_{i\in\mathcal{V}_\eta}$ and weighted gain matrices $\{G_{i,\eta}/N_\eta\}_{i\in\mathcal{V}_\eta}$ in row and column, respectively.
- Control gain matrix K satisfies Assumption 1.
- Communication parameter $\alpha_\eta \in (0, 2/\lambda_{\max}(\mathcal{L}_\eta))$.

IV. Performance Analysis

A. Upper bounds of estimation error and tracking deviation

The following theorem provides upper bounds of the state estimation error and the trajectory tracking deviation respectively, i.e., $\bar{h}_e(t)$ and $\bar{h}_c(t)$, whose definitions are provided in [6] due to space constraint.

Theorem 1: Consider the system and Algorithms 1–2 satisfying Assumptions 1–4 with parameters $\{G_{i,\eta}\}$, K, and $\{\alpha_\eta\}$ designed as in Lemma 2. The estimation error of each active sensor and the trajectory tracking deviation are both upper bounded for any task η , i.e., for any $t \in [t_{\eta-1}+1, t_\eta]$,

$$\|\hat{x}_i(t) - x(t)\| \leq \bar{h}_e(t), \quad \|x(t) - r(t)\| \leq \bar{h}_c(t), \qquad (8)$$

if communication parameter L_η is subject to

$$L_\eta \geq l_\eta \qquad (9)$$

where $l_\eta > 0$ is a scalar specified in [6].

When there is no uncertainty in system dynamics and measurements, the estimation and tracking performance of the architecture is illustrated in the following corollary, whose proof follows from Theorem 1.

Corollary 1: Under the same conditions as in Theorem 1, if the system is uncertainty-free, i.e., $q_w = q_v = 0$, the following result holds

$$\lim_{t\to\infty} \|\hat{x}_i(t) - x(t)\| = 0, \quad \lim_{t\to\infty} \|x(t) - r(t)\| = 0,$$

provided that the task is not switched.

From Theorem 1, a design of $\{h_e(t)\}$ and $\{h_c(t)\}$ for task-switching condition (7) is provided in the following corollary.

Corollary 2: For task η with $t \in [t_{\eta-1}+1, t_\eta]$, let

$$h_e(t) = \|D_\eta\| \, \bar{h}_e(t), \quad h_c(t) = \|D_\eta\| \, \bar{h}_c(t),$$

then under the same conditions as in Theorem 1 sequences $\{h_e(t)\}_{t=t_{\eta-1}+1}^{t_\eta}$ and $\{h_c(t)\}_{t=t_{\eta-1}+1}^{t_\eta}$ satisfy the requirement in Lemma 1.

Under Assumption 3, the robot is able to compute $\{h_e(t)\}_{t=t_{\eta-1}+1}^{t_\eta}$ and $\{h_c(t)\}_{t=t_{\eta-1}+1}^{t_\eta}$ online such that they can be used in Algorithm 1 for the event-triggered task switching.

B. Conditions of task accomplishment

In the following theorem we provide a sufficient condition to ensure the accomplishment of a task. Moreover, we study the running time for each task.

Theorem 2: Suppose task $\eta-1$ is accomplished, then consider Algorithms 1–2 for task η. Under the same conditions as in Corollary 2, if $\lim_{l\to\infty} \|D_\eta r_{\eta-1,\eta}(l) - c_\eta\| = 0$, and

$$R_\eta > \tau > 0 \qquad (10)$$

then task η will be accomplished for sure, where τ is specified in [6]. Furthermore, the running time for task η is upper bounded by $t_{\min,\eta}$, where

$$t_{\min,\eta} = \min_{\Delta t \geq \mathbb{T}_\eta} \Delta t, \quad \text{s.t.} \max_{l\in[\Delta t - \mathbb{T}_\eta, \Delta t)} \{\phi(\eta, l)\} \leq R_\eta, \qquad (11)$$

where $\phi(\eta, l) = h_c(t_{\eta-1}+l) + \|D_\eta r_{\eta-1,\eta}(l) - c_\eta\|$.

Theorem 2 shows that it is feasible to obtain an upper bound of the task running time against system uncertainties, and the initial estimation error and tracking deviation. This bound can be used to evaluate the efficiency of the architecture and the generation of reference trajectories. Although optimization problem (11) could be non-convex, it can be solved offline by using some existing algorithms, such as enumeration methods or heuristic optimization algorithms.

V. Numerical Simulations

In this section, we consider the motivating example in Fig. 3 on robot navigation over DSNs. The robot's state is four-dimensional, and it consists of longitude and latitude positions and the corresponding velocities. The parameter matrices in (1) are assumed to be $A = \begin{pmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{pmatrix}$, and $B = \begin{pmatrix} 0 & 0 \\ T & 0 \\ 0 & 0 \\ 0 & T \end{pmatrix}$, where $T = 0.01$. Each active sensor $i = 1, 2, \ldots, 12$ is assumed to have the following measurement matrix $C_i = (1, 0, 0, 0)$ if i is odd, otherwise $C_i = (0, 0, 1, 0)$. Assume each element of the process uncertainty follows a uniform distribution in [-0.005.0.005]. The measurement uncertainty follows a uniform distribution in [-0.01.0.01]. The initial state of the robot $x_0 = (100, -0.1, 20, -0.06)^\mathsf{T}$. Each element of the initial estimation error of each sensor follows a uniform distribution in [-5,5]. The centers of the three targeted sets are $c_1 = (50, 5)^\mathsf{T}, c_2 = (20, 50)^\mathsf{T}, c_3 = (80, 60)^\mathsf{T}$ with radius $R_1 = 5, R_2 = 10$, and $R_3 = 15$ respectively. The sets are used to represent the desired positions. With these parameters and constraint (2), we use cubic spline interpolation to generate three reference trajectories connecting the centers of the three targeted sets, respectively. The lengths of the three trajectories are 501, 301, and 601 respectively.

Given each active DSN, choose parameters $\{G_{i,\eta}\}$ and K, $\eta = 1, 2, 3$, for Algorithms 1–2 as in Lemma 2, such that the eigenvalues of $A - G_\eta \tilde{C}_\eta$ and $A + BK$ are placed at $(-0.2, -0.1, 0.1, 0.2)^\mathsf{T}$ and $(-0.1, -0.1, 0.1, 0.1)^\mathsf{T}$, respectively. Choose parameter $\alpha_\eta = 2/(\lambda_2(\mathcal{L}_\eta) + \lambda_{\max}(\mathcal{L}_\eta))$. The desired dwelling time for each targeted set is $\mathbb{T}_\eta = 2$. Communication parameter $L_\eta = 5$. The design of $\{h_e(t)\}$ and $\{h_c(t)\}$ is as in Corollary 2 and Algorithm 3 in [6].

We conduct a Monte Carlo experiment with 100 runs. Under the above setting, we run Algorithms 1–2 and obtain Figs. 3–4. We randomly choose one realization to show the performance of the robot in tracking the reference trajectory (position) in Fig. 3. It shows the robot is able to reach each targeted set successfully and switch to the next task sequentially. In Fig. 4, the dynamics of event-triggered parameter $f(t)$ in Algorithm 1 and the task-switching points are shown. The decrease of $f(t)$ is due to the performance improvement for state estimation and trajectory tracking as time goes on, while the fluctuation of $f(t)$ is resulted from task switching.

Next, we consider task one within time interval [1,400] under the same setting as in the first case to compare the navigation performance of Algorithms 1–2 with some other algorithms. The centralized Kalman filter (CKF) and the CSGE [12] are considered here, and they are equipped with our feedback controller in Algorithm 1 and then abbreviated by CKFFC and CSGEFC, respectively. Moreover, we study navigation performance by using the reference input. We compare these algorithms based on the average navigation error (trajectory-tracking deviation): $\epsilon(t) = \frac{1}{100}\sum_{j=1}^{100} \left\|x^j(t) - r(t)\right\|$. The comparison result is provided in Fig. 5, which shows that the navigation errors of

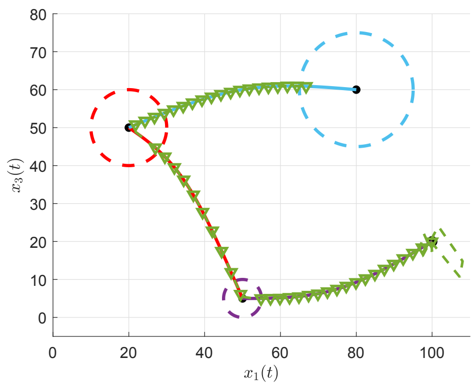


Fig. 3. Reference position trajectory tracking and task switching of Algorithms 1–2. In the transient time period, since the estimate is not accurate, the estimate-based control input does not drive the robot to the reference position trajectory. As time goes on, the robot is able to converge to the reference trajectory and accomplish the three tasks sequentially.

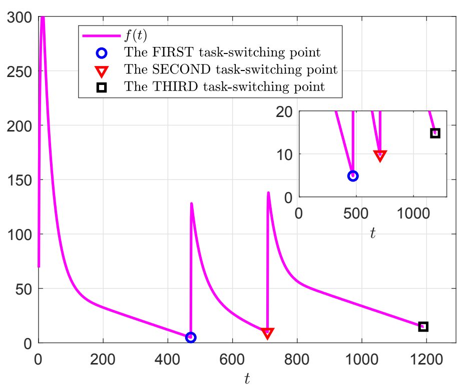


Fig. 4. The dynamics of event-triggered parameter $f(t)$ and the task-switching points. In each task, parameter $f(t)$ decreases until the current task is accomplished. From one task to another, the sharp increase of $f(t)$ is due to the switching of the reference trajectories.

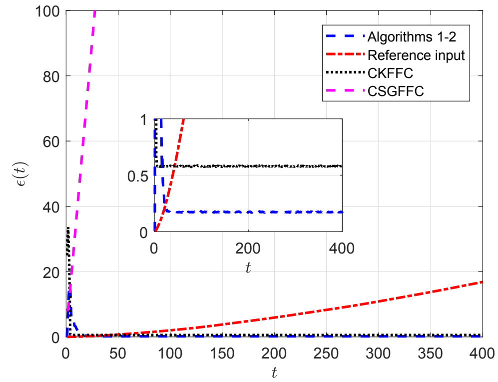


Fig. 5. Comparison of navigation errors of several algorithms for task one.

Algorithms 1–2 and CKFFC are both tending to small neighborhoods of zero. Meanwhile, Algorithms 1–2 outperforms the other three algorithms in the considered scenario. Note that the CKF is not optimal in this case, since the uncertainties here follow a uniform distribution but not a normal distribution. Moreover, Fig. 5 illustrates that feedback control outperforms the reference control for trajectory tracking in uncertain environments.

VI. Conclusions

We studied how to navigate a mobile robot over active distributed sensor networks. We proposed a new task-switching navigation architecture consisting of a feedback controller and a distributed estimator. We found two time-varying sequences to bound the state estimation error and trajectory tracking deviation in the architecture, respectively. Moreover, a sufficient condition for the accomplishment of each task was established and an upper bound of the running time of each task was obtained. In the future work, it is interesting to extend the results to nonlinear systems and study the optimal sensor selection method. In addition, it would be interested to study multiple robots utilizing the same network of sensors simultaneously by considering the traffic on these sensors, inter-robot safety conditions as well as communications and exogenous actors.

References

[1] H. Chung, C. Hou, and Y. Chen, "Indoor intelligent mobile robot localization using fuzzy compensation and Kalman filter to fuse the data of gyroscope and magnetometer," *IEEE Trans. on industrial electronics*, vol. 62, no. 10, pp. 6436–6447, 2015.

[2] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis, "A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 829–846, 2018.

[3] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237–267, 2002.

[4] G. Battistelli and L. Chisci, "Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability," *Automatica*, vol. 50, no. 3, pp. 707–718, 2014.

[5] S. Park and N. C. Martins, "Design of distributed lti observers for state omniscience," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 561–576, 2016.

[6] X. He, E. Hashemi, and K. H. Johansson, "Navigating a mobile robot using switching distributed sensor networks," *arXiv preprint arXiv:2106.13529*, 2021.

[7] P. N. Guerra, P. J. Alsina, A. A. Medeiros, and A. P. Araújo, "Linear modelling and identification of a mobile robot with differential drive," in *International Conference on Informatics in Control, Automation and Robotics*, pp. 263–269, 2004.

[8] C. Belta and S. Sadraddini, "Formal methods for control synthesis: An optimization perspective," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 115–140, 2019.

[9] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Review*, vol. 59, no. 4, pp. 849–904, 2017.

[10] A. d'Onofrio, *Bounded Noises in Physics, Biology, and Engineering*. Springer, 2013.

[11] D. Marelli, T. Sui, and M. Fu, "Distributed Kalman estimation with decoupled local filters," *Automatica*, vol. 130, p. 109724, 2021.

[12] U. A. Khan and A. Jadbabaie, "Collaborative scalar-gain estimators for potentially unstable social dynamics with limited communication," *Automatica*, vol. 50, no. 7, pp. 1909–1914, 2014.