IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Multi-Robot Active Sensing and Environmental Model Learning With Distributed Gaussian Process

Dohyun Jang[1], Jaehyun Yoo[2], Clark Youngdong Son[1], Dabin Kim[1], and H. Jin Kim[1]

***Abstract*— This paper deals with the problem of multiple robots working together to explore and gather at the global maximum of the unknown field. Given noisy sensor measurements obtained at the location of robots with no prior knowledge about the environmental map, Gaussian process regression can be an efficient solution to construct a map that represents spatial information with confidence intervals. However, because the conventional Gaussian process algorithm operates in a centralized manner, it is difficult to process information coming from multiple distributed sensors in real-time. In this work, we propose a multi-robot exploration algorithm that deals with the following challenges: i) distributed environmental map construction using networked sensing platforms; ii) online learning using successive measurements suitable for a multi-robot team; iii) multi-agent coordination to discover the highest peak of an unknown environmental field with collision avoidance. We demonstrate the effectiveness of our algorithm via simulation and a topographic survey experiment with multiple UAVs.**

I. Introduction

Multi-robot systems can be better specialized than a single robot for missions in large areas such as crop monitoring, observing climate changes, and terrain surveying [1]. These missions can be viewed as environmental process estimation problems in which each robot measures spatio-temporal data on its own. As such, techniques for multiple robots to move around and acquire local information for constructing a global environmental map are referred to as robotic sensor networks [2].

In robotic sensor networks, robots move to optimal locations where most valuable data can be obtained, and these data are used to generate an environmental map. We need to learn an environmental model from the collected data and estimate the information gain to decide which location to investigate further.

Another challenge in multi-robot systems is the limitation of network resources such as communication distance, transmission throughput, and channel bandwidth [3], [4]. Given these issues, network control systems (NCS) have been developed to facilitate the cooperative work by using mutually agreed protocols for data communication of a large number of robots [5], [6].

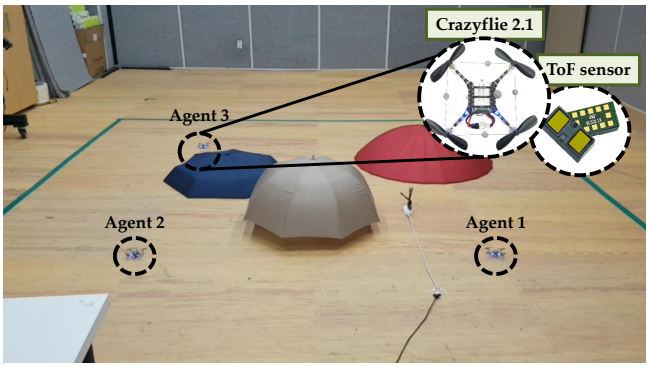


Fig. 1: Topographic survey experiments with multiple UAVs

In this paper, we present a distributed multi-robot exploration method for joint sensing and learning of an unknown environmental process with the following challenges: i) distributed learning for an environmental process and communication protocol in networked control systems, ii) online update of an environmental process model with newly acquired sensory data for multi-robot systems, and iii) multi-robot coordination for improving the process estimation performance.

A. Related work

Gaussian process (GP) regression [7] can be used to construct an environmental process from data [8], [9]. It derives a spatial relationship between sampled data by using a kernel function and performs Bayesian inference for prediction at an unknown location. An online version of GP [10], which is extended from typical batch learning, would be more suitable for real-time robot learning scenarios [11].

However, most online GP approaches have been designed for centralized systems, making it difficult to apply them for a multi-robot NCS subject to the limitation of network resources (e.g., transmission power, throughput, or channel bandwidth). Even though relay communication might be a solution for large-scale data transition [12], [13], it takes too much time and bandwidth to adapt rapidly changing environment in multi-robot NCS.

In [14], distributed GP regression is introduced for multi-agent systems, in which a finite-dimensional GP estimator is designed by using Karhunen–Loève (KL) kernel expansion. It is more flexible on a network scale because only a connected graph is needed where each node is connected to

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2020-2017-0-01637) supervised by the IITP(Institute for Information & communications Technology Promotion)

[1]Dohyun Jang, Clark Youngdong Son, Dabin Kim, and H. Jin Kim are with Department of Mechanical and Aerospace Engineering, Seoul National University (SNU) and Automation and Systems Research Institute (ASRI), Seoul, South Korea (email:{dohyun,clark.y.d.son,dabin404,hjinkim}@snu.ac.kr)

[2]Jaehyun Yoo is with the Department of Electrical, Electronic and Control Engineering, Hankyong National University, Anseong, South Korea (email:jhyoo@hknu.ac.kr)

the other nodes through a sequence of edges (i.e. path), not a centralized network where all agents are directly connected to a central server. In contrast to the decentralized GP's local estimate, the distributed GP yields a global estimate as if using the sensory data of all agents by exchanging estimator information with neighbors. However, this study considers sensor networks only, not moving robot platforms. Thus, the sensor position is fixed, and data cannot be obtained where the sensor is not installed. The main difference with our work lies in the fact that we can update the GP estimate using newly measured sensory data from a multi-robot team.

We are interested in robotic sensor networks that could enable active sensing and constructing the environmental model using mobile robots. [15] represents informative planning for autonomous robot with online sparse GP, which takes advantage of a subset of data. In contrast to our work, it considered only single-agent exploration. In [16], a multi-robot sensor coverage problem is addressed with a mixture of the Gaussian process. These papers employ multi-robot systems and GP but concentrate only on the optimal sensor placement objectives without consideration of collision avoidance.

The idea of combining GP and multi-robot systems has already been explored in several works. [17] presents a decentralized multi-agent exploration with online GP, focusing on multi-agent coordination and physical process construction. In [18], Gaussian radial basis functions and multi-agent systems are utilized to discover peak, considering the communication distance. In contrast to our study, both [17] and [18] only address the local-estimate-based exploration, not the global-estimate-based. With local estimates for multi-robot control, each agent would follow an inefficient path or converge to the local peaks. [20] developed multi-UAS planning with a decentralized GP fusion algorithm, which takes a different approach from our works in implementing a distributed GP. It has an advantage of communication efficiency; however, this technique does not guarantee the global estimate's performance.

B. Our Contribution

This paper focuses on three main contributions to multi-robot exploration in the networked control system.

- We present a distributed GP algorithm with global mean and variance estimation through Karhunen–Loève expansion and an average consensus protocol [14].
- We expand the distributed GP algorithm to be suitable for continuous data collection to enable online GP learning for mobile robots.
- To find the location of the peak value of a scalar function, which has been considered in [18], [23], we propose the maximum variance exploration and maximum mean exploitation for individual agents. For safe and efficient exploration, we apply a collision avoidance and coordination algorithm.

We perform a multi-robot exploration simulation in a virtual environment and conduct a topographic survey experiment using multiple unmanned aerial vehicles (UAVs) with a laser rangefinder. The outline of this paper is as follows. Section II formally states distributed multi-robot systems and briefly introduces a background. Section III summarizes distributed GP. Section IV combines distributed GP and multi-robot coordination. Simulation and real-time experiments are presented in Section V. Section VI concludes the paper.

II. PRELIMINARY AND SYSTEM MODELING

A. Problem Statement

Multiple robots (e.g., ground vehicles or UAVs) explore an unknown environmental process in 3-D space with onboard sensors. Each agent estimates environmental information across an unknown area that has not yet been explored by using both own measurements and shared data received from neighbors. The robot determines its next location to move based on the prediction by the distributed GP, where a search for a place of high uncertainty is prioritized. We consider the following setting:

- The exploration area is finite, and all agents know the boundary of the exploration area. The environmental process is time-invariant.
- Each agent has its localization and navigation capabilities in the global coordinate system.
- Each agent can only communicate directly with agents within the communication range. Communication between the two agents is bidirectional.

B. Multi-Robot System

We consider N agents to explore the environment. All agents have the same dynamics model. Each robot has the 3-D position state $\mathbf{x}_i \in \mathbb{R}^3$ for $i = 1, \cdots, N$ and takes the measurement of environmental process $y_i \in \mathbb{R}$, which is corrupted by the white Gaussian noise $\nu_i \sim \mathcal{N}(0, \sigma_\nu^2)$, having the following relationship:

$$y_i \;=\; f(\mathbf{x}_i) + \nu_i, \tag{1}$$

where the measurement model $f(\cdot)$ will be estimated by distributed GP regression algorithm.

During exploration, robots estimate environmental model by using both measured and shared data obtained from neighbors. Each robot can only communicate with adjacent robots directly within a certain distance. The communication network of N robots is defined based on graph theory. Let $\mathcal{G}(k) = \{\mathcal{V}, \mathcal{E}(k)\}$ be an undirected graph of order N with the non-empty set of nodes $\mathcal{V} = \{v_i |\ i \in \mathcal{N}\}$ and the set of edges $\mathcal{E}(k) = \{(v_i, v_j) |\ i, j \in \mathcal{N}, ||\mathbf{x}_i - \mathbf{x}_j|| < l_{comm}\}$ at k, where $\mathcal{N} = \{1, 2, \cdots, N\}$ is the index set of nodes. (v_i, v_j) is the communication channel from the robot i to j, and l_{comm} is the communication range of the robot. $\mathcal{N}_i(k) = \{j |\ (v_j, v_i) \in \mathcal{E}(k), j \in \mathcal{N}\}$ is the set of neighbors of the robot i.

C. Average Consensus Algorithm

Each agent can receive data only from neighbor agents due to the limitations of the communication network. Against the communication constraint, the consensus algorithm enables multiple agents to operate on the same protocol so that each

local estimate converges to the global estimate. More details for obtaining a global estimate are addressed in Section III.

We define the following dynamics with the state matrix $\{X_i\}_{i=1}^N$ and the consensus protocol $\{U_i\}_{i=1}^N$:

$$X_i((k+1)T) = X_i(kT) + U_i(kT). \tag{2}$$

where $k \in \mathbb{Z}_{\geq 0}$, and T is the sampling time. We replace all "(kT)" by "(k)" for brevity. The *average consensus* for these N state matrices is satisfied when the following two conditions are met [6]:

$$\lim_{k\to\infty} ||X_i(k) - X_j(k)|| = 0, \quad \forall i,j \in \mathcal{N} \tag{3}$$

and

$$\sum_{i\in\mathcal{N}} X_i(k) = \bar{X}(k_0), \quad \forall k \in (k_0, \infty). \tag{4}$$

The variable $\bar{X}(k)$ is the average matrix of all state matrices X_i at k. If only (3) is satisfied, *consensus* is achieved.

When the graph $\mathcal{G}(k)$ is connected, the protocol for the system (2) to achieve the average consensus is as follows [5]:

$$U_i(k) = -\frac{1}{|\mathcal{N}_i(k)|} \sum_{j\in\mathcal{N}_i(k)} \gamma(X_i(k) - X_j(k)) \tag{5}$$

for agent $i \in \mathcal{N}$. $|\mathcal{N}_i(k)|$ is the cardinality of $\mathcal{N}_i(k)$ and $\gamma \in \mathbb{R}$ is the parameter for convergence rate. We will show how the multi-agent team performs distributed Gaussian process regression in the networked control systems using this average consensus protocol.

III. DISTRIBUTED GAUSSIAN PROCESS

A. Gaussian Process Regression

GP regression is popular in modeling spatial phenomena. This data-driven non-parametric learning can provide probabilistic inferences over the entire space, taking into account joint Gaussian probability distribution between the sample data [7]. In (1), the unknown measurement model $f(\cdot)$ is assumed to be zero-mean Gaussian such that

$$f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')). \tag{6}$$

The Gaussian kernel function $k(\mathbf{x}, \mathbf{x}')$ is defined as

$$k(\mathbf{x}, \mathbf{x}') = \sigma_s^2 \exp(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \Sigma^{-1} (\mathbf{x} - \mathbf{x}')), \tag{7}$$

where σ_s^2 is the signal variance of $f(\mathbf{x})$, and Σ is the length scale that determines how fast the correlation between data points decreases. Typically, the hyper parameters σ_s^2 and Σ can be learned by the evidence maximization algorithm [7].

We define the training input data $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$, which contains all locations of each agent and the training target $\mathbf{y} = [y_1 \cdots y_N]^T$, which is sensory data vector corresponding to $\mathbf{X}$. When the test location $\mathbf{x}_*$ is given, the posteriori distribution of $f(\mathbf{x}_*)$ is derived as follows:

$$p(f(\mathbf{x}_*)|\mathbf{X}, \mathbf{y}, \mathbf{x}_*) \sim \mathcal{N}(\hat{f}(\mathbf{x}_*), \Sigma(\mathbf{x}_*)) \tag{8}$$

where

$$\hat{f}(\mathbf{x}_*) = K(\mathbf{x}_*, \mathbf{X})(K(\mathbf{X}, \mathbf{X}) + \sigma_v^2 I)^{-1}\mathbf{y}, \tag{9a}$$

$$\begin{aligned}\Sigma(\mathbf{x}_*) = {} & k(\mathbf{x}_*, \mathbf{x}_*) \\ & - K(\mathbf{x}_*, \mathbf{X})(K(\mathbf{X}, \mathbf{X}) + \sigma_v^2 I)^{-1} K(\mathbf{X}, \mathbf{x}_*).\end{aligned} \tag{9b}$$

The i,j-th element of the kernel matrix $K(\cdot,\cdot)$ is calculated by (7).

B. Karhunen–Loève (KL) Kernel Expansion

The usual kernel-based GP uses the correlation between all the sample data, making it difficult to apply to a distributed GP where each agent cannot access all the sample data. Therefore, a new kernel transformation is needed. The kernel (7) can be expanded in terms of eigenfunctions ϕ_e and related eigenvalues λ_e [21]. They are defined by

$$\lambda_e \phi_e(\mathbf{x}) = \int_\chi k(\mathbf{x}, \mathbf{x}') \phi_e(\mathbf{x}') d\mu(\mathbf{x}'), \tag{10}$$

$$k(\mathbf{x}, \mathbf{x}') = \sum_{e=1}^{+\infty} \lambda_e \phi_e(\mathbf{x}) \phi_e(\mathbf{x}'). \tag{11}$$

Defining the kernel expansion in a closed form, in general, is difficult but the Gaussian kernel expansion has already been studied. The Gaussian kernel expansion is obtained via Hermite polynomials, as mentioned in [19]. Then, the measurement model f in (1) is reformulated as

$$\begin{aligned} f(\mathbf{x}) &= \sum_{e=1}^{E} a_e \phi_e(\mathbf{x}) + \sum_{e=E+1}^{+\infty} a_e \phi_e(\mathbf{x}) \\ &= f_E(\mathbf{x}) + \sum_{e=E+1}^{+\infty} a_e \phi_e(\mathbf{x}) \end{aligned} \tag{12}$$

where

$$a_e \sim \mathcal{N}(0, \lambda e), \quad e = 1, 2, \cdots \tag{13}$$

$f_E(\mathbf{x})$ is the E-dimensional approximation of $f(\mathbf{x})$. As proven in [19], the optimal E-dimensional linear models can be expressed by a combination of the first E-kernel eigenfunctions as the learning data set size grows to infinity.

C. Multi-Agent Distributed Gaussian Process

In order to apply the expanded kernel to distributed systems, a transformation of function (9a) to finite dimensions is required. Combining (9a) and (12), the Gaussian process E-dimensional estimator is designed as follows [14]:

$$\hat{f}_E(\mathbf{x}) \;:=\; \Phi^T(\mathbf{x}) H_E \mathbf{y} \tag{14}$$

where

$$\Phi(\mathbf{x}) := \left[\phi_1(\mathbf{x}), \cdots, \phi_E(\mathbf{x})\right]^T \tag{15a}$$

$$H_E := \left(\frac{G^T G}{Nm} + \frac{\sigma_\nu^2}{Nm}\Lambda_E^{-1}\right)^{-1} \frac{G^T}{Nm} \tag{15b}$$

$$G := \left[\Phi(\mathbf{x}_1), \cdots, \Phi(\mathbf{x}_N)\right]^T \tag{15c}$$

Λ_E is the diagonal matrix of kernel eigenvalues and m represents the number of data acquired by each agent. For convenience of explanation, this subsection assumes that all agent positions are fixed so that m is 1.

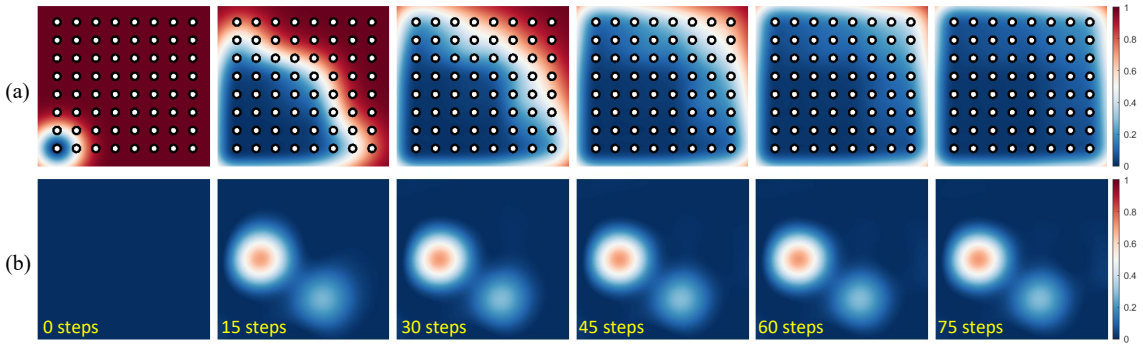


Fig. 2: The process of the environmental model construction performed by 64 stationary sensors with distributed Gaussian process regression. It shows (a) the uncertainty propagation and (b) the change of the GP estimate with time from $k = 0$ to $k = 75$ in order from the left figure. The environmental model is presented in Fig 3. The communication range is equal to the distance between two adjacent agents. All results are obtained by agent $\#1$ located at the bottom-left corner. In addition to the results of agent $\#1$, the results of all the other agents converge to the same.

In networked control system, each agent cannot obtain G and $\mathbf{y}$ in (14) without a fully connected network. To replace G and $\mathbf{y}$, we decompose the related terms in (15b) as follows:

$$\frac{G^T G}{Nm} = \frac{1}{N}\sum_{i=1}^{N}\Phi(\mathbf{x}_i)\Phi^T(\mathbf{x}_i) = \frac{1}{N}\sum_{i=1}^{N}\alpha_i(k_0) \tag{16a}$$

$$\frac{G^T \mathbf{y}}{Nm} = \frac{1}{N}\sum_{i=1}^{N}\Phi(\mathbf{x}_i)y_i = \frac{1}{N}\sum_{i=1}^{N}\beta_i(k_0) \tag{16b}$$

Now (14) is reformulated in the following distributed form:

$$\hat{f}_{E,i}(\mathbf{x}) \;:=\; \Phi^T(\mathbf{x})\left(\alpha_i(k) + \frac{\sigma_\nu^2}{N}\Lambda_E^{-1}\right)^{-1}\beta_i(k) \tag{17}$$

We apply the average consensus algorithm in Section II. *D* to $\alpha_i(k)$ and $\beta_i(k)$ for $k \geq k_0$:

$$\begin{aligned}\alpha_i(k+1) &= \alpha_i(k) + \Delta\alpha_i(k)\\ \beta_i(k+1) &= \beta_i(k) + \Delta\beta_i(k)\end{aligned} \tag{18}$$

where

$$\begin{aligned}\Delta\alpha_i(k) &= -\sum_{j\in\mathcal{N}_i}\gamma(\alpha_i(k) - \alpha_j(k))\\ \Delta\beta_i(k) &= -\sum_{j\in\mathcal{N}_i}\gamma(\beta_i(k) - \beta_j(k)).\end{aligned} \tag{19}$$

As the results of average consensus, (17) converges to (14).

In the same way, the distributed Gaussian process approximation for $\Sigma(\mathbf{x})$ in (9b) is defined as

$$\Sigma_E(\mathbf{x}) \;:=\; k(\mathbf{x},\mathbf{x}) - \Phi^T(\mathbf{x})H_E G\Lambda_E\Phi(\mathbf{x}) \tag{20}$$

$$\begin{aligned}\Sigma_{E,i}(\mathbf{x}) :=\; & k(\mathbf{x},\mathbf{x})\\ & -\Phi^T(\mathbf{x})\left(\alpha_i + \frac{\sigma_\nu^2}{N}\Lambda_E^{-1}\right)^{-1}\alpha_i\Lambda_E\Phi(\mathbf{x})\end{aligned} \tag{21}$$

According to (9) and (14), the computational complexity of the distributed system is $O(E^3)$, whereas that of the centralized system is $O((Nm)^3)$ because of matrix inversion [14]. Thus, the distributed system is more scalable to the number of agents and the amount of sensory data where typically $E \ll Nm$.

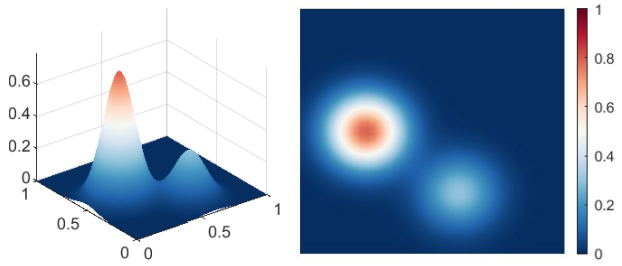


Fig. 3: Bi-modal environmental phenomenon for simulation. (left) 3D perspective view. (right) top-down view.

Fig. 2 shows distributed GP example using 64 stationary agents for unknown environmental model in Fig. 3. The overall uncertainty for the workspace decreases over time, and the GP mean estimate converges to the original model.

D. Continuous Information Acquisition and Fusion by Moving Agent

The results in Section III-C only considered the stationary sensors with the initially measured data set. When the m-th sensor measurement $y_i(k_{m-1})$ is obtained on $\mathbf{x}_i(k_{m-1})$, (16a) becomes following equation:

$$\begin{aligned}\frac{G^T G}{Nm} &= \frac{1}{Nm}\sum_{i=1}^{N}\sum_{j=0}^{m-1}\Phi(\mathbf{x}_i(k_j))\Phi^T(\mathbf{x}_i(k_j))\\ &= \frac{1}{N}\sum_{i=1}^{N}\alpha_i(k),\end{aligned} \tag{22a}$$

$$\begin{aligned}\frac{G^T \mathbf{y}}{Nm} &= \frac{1}{Nm}\sum_{i=1}^{N}\sum_{j=0}^{m-1}\Phi(\mathbf{x}_i(k_j))y_i(k_j)\\ &= \frac{1}{N}\sum_{i=1}^{N}\beta_i(k).\end{aligned} \tag{22b}$$

In (16), the initial $\{\alpha_i(k_0)\}_{i=1}^N$ and $\{\beta_i(k_0)\}_{i=1}^N$ were only used for the average consensus algorithm. However, if each agent obtain new sensor measurement while $\{\alpha_i(k)\}_{i=1}^N$ and $\{\beta_i(k)\}_{i=1}^N$ are approaching the consensus, $\{\alpha_i(k)\}_{i=1}^N$ and

Algorithm 1 Multi-Robot Exploration for agent #*i*

$\alpha_i(k < k_0) \leftarrow 0, \beta_i(k < k_0) \leftarrow 0$
$k \leftarrow k_0, m \leftarrow 1$
while *True* **do**
  */*Sensing*/*
  if mod(*k*,*SensingPeriod*) $= 0$ **then**
    $y_i(k) \leftarrow f(\mathbf{x}_i(k)) + \nu$
    $\alpha_i(k) \leftarrow \frac{m-1}{m}\alpha_i(k-1) + \frac{1}{m}\Phi(\mathbf{x}_i(k))\Phi^T(\mathbf{x}_i(k))$
    $\beta_i(k) \leftarrow \frac{m-1}{m}\beta_i(k-1) + \frac{1}{m}\Phi(\mathbf{x}_i(k))y_i(k)$
    $m \leftarrow m + 1$
  end
  */*Communication*/*
  $\Delta\alpha_i(k) \leftarrow -\sum_{j\in\mathcal{N}_i}\gamma(\alpha_i(k) - \alpha_j(k))$
  $\Delta\beta_i(k) \leftarrow -\sum_{j\in\mathcal{N}_i}\gamma(\beta_i(k) - \beta_j(k))$
  $\alpha_i(k+1) \leftarrow \alpha_i(k) + \Delta\alpha_i(k)$
  $\beta_i(k+1) \leftarrow \beta_i(k) + \Delta\beta_i(k)$
  */*Multi-Robot Coordination* $(Algorithm2)$**/*
  $\mathbf{x}_i(k+1) \leftarrow \pi(\mathbf{x}_i(k), \alpha_i(k), \beta_i(k))$
  $k \leftarrow k + 1$
end

$\{\beta_i(k)\}_{i=1}^N$ imply to be recalculated to include new data, and it has to restart the consensus process from the beginning as shown in (22). Thus, we propose the following continuous information acquisition algorithm.

Let assumes that the sensor data acquisition rates of all agents are identical for convenience. The transition model of $\alpha_i(k)$ and $\beta_i(k)$ at k_{m-1} are represented as follows:

$$\begin{aligned}\alpha_i(k) &= \tfrac{m-1}{m}\alpha_i(k-1) + \tfrac{1}{m}\Phi(\mathbf{x}_i(k))\Phi^T(\mathbf{x}_i(k)) \\ \beta_i(k) &= \tfrac{m-1}{m}\beta_i(k-1) + \tfrac{1}{m}\Phi(\mathbf{x}_i(k))y_i(k),\end{aligned} \tag{23}$$

where $\alpha_i(k < k_0) = 0$ and $\beta_i(k < k_0) = 0$. If $\{\alpha_i(k)\}_{i=1}^N$ and $\{\beta_i(k)\}_{i=1}^N$ approach to the leftmost terms in (22a) and (22b) respectively, (17) and (21) converge to (14) and (20) respectively.

Theorem 1: By updating $\{\alpha_i(k)\}_{i=1}^N$ and $\{\beta_i(k)\}_{i=1}^N$ using (18) and (23), new data are naturally blended with the existing $\alpha_i(k-1)$ and $\beta_i(k-1)$, so that $\alpha_i(k)$ and $\beta_i(k)$ move towards the leftmost terms in (22a) and (22b), respectively, in a distributed manner (Algorithm 1).

Proof: See The Appendix. ■

IV. MULTI-AGENT ACTIVE SENSING AND CONTROL

A. Exploration and Exploitation

In the previous section, we introduced the distributed GP to construct an unknown environmental model by letting multiple agents explore. In this section, we establish a multi-robot exploration strategy to improve GP performance with additional data collection and exploitation strategies to improve problems when focusing only on exploration.

To make up for the lack of information in the initial location, each robot would move in the direction of the region

Algorithm 2 Multi-Robot Control Policy π for agent #*i*

input : $\mathbf{x}_i, \alpha_i, \beta_i$
output : $\mathbf{x}_{i,next}$
1: $l \leftarrow l_{init}$
2: *ExplorationFlag* $\leftarrow$ *True*
3: $\Theta_{init} \leftarrow [0, 2\pi)$
4: **while** *True* **do**
5: */*Boundary condition*/*
6: $\Theta_{bc} \leftarrow \{\theta_{bc} \in \Theta_{init} |\ (l\angle\theta_{bc} + \mathbf{x}_i) \in X_{map}\}$
7: */*Collision Avoidance condition*/*
8: $\Theta_{ca} \leftarrow \{\theta_{ca} \in \Theta_{init} |\ ||(l_{init}\angle\theta_{ca} + \mathbf{x}_i) - \mathbf{x}_j|| > l_{ca},$
9: $\forall j \in \mathcal{N}_i\}$
10: */*Motion Candidates*/*
11: $\Theta_c \leftarrow \Theta_{bc} \cap \Theta_{ca}$
12: **if** $\max_{\theta_c\in\Theta_c} \Sigma_{E,i}(l\angle\theta_c + \mathbf{x}_i) > \sigma^2_{\text{threshold}}$ **then**
13: **break**
14: **else**
15: $l \leftarrow l + l_{init}$
16: **if** $l > l_{\text{threshold}}$ **then**
17: *ExplorationFlag* $\leftarrow$ *False*
18: **break**
19: **end**
20: **end**
21: **end**
22: **if** *ExplorationFlag* **then**
23: */*Exploration*/*
24: $\theta_{d,i} \leftarrow \arg\max_{\theta_c\in\Theta_c}[w_i(l\angle\theta_c + \mathbf{x}_i) \times \Sigma_{E,i}(l\angle\theta_c + \mathbf{x}_i)]$
25: **else**
26: */*Exploitation*/*
27: $\theta_{d,i} \leftarrow \arg\max_{\theta_c\in\Theta_c}[\hat{f}_{E,i}(l\angle\theta_c + \mathbf{x}_i) + \eta\Sigma_{E,i}(l\angle\theta_c + \mathbf{x}_i)]$
28: **end**
29: $\mathbf{x}_{i,next} \leftarrow l_{init}\angle\theta_{d,i}$

where the uncertainty of GP estimation is high (exploration). Each robot sets a searching area like (24) based on its current location $\mathbf{x}_i$ and calculates the distributed GP variance (21).

$$X_c = \{\mathbf{x}_c = l\angle\theta_c + \mathbf{x}_i |\ \theta_c \in [0, 2\pi)\} \tag{24}$$

where l is the search range. If the uncertainty around the robot is lower than a parameter $\sigma^2_{\text{threshold}}$, the search for nearby areas of the robot is not likely to be effective for improving the overall amount of information. In this case, the robot should broaden the search area and examine the GP uncertainty over a wider area (line 15 of Algorithm 2). If there is a region where the uncertainty exceeds $\sigma^2_{\text{threshold}}$, the robot moves toward the direction decided by the lines 12-13,23-24,29 of Algorithm 2. If the robot cannot find an area having the uncertainty above a certain threshold after trying expanding the search area, it enters the exploitation process (lines 17-20,27-28 of Algorithm 2).

The goal of the robots is to find the highest value for the environmental process. GP increases the uncertainty as the test location moves away from the data acquisition location. Therefore, exploration techniques that chase only uncertainty

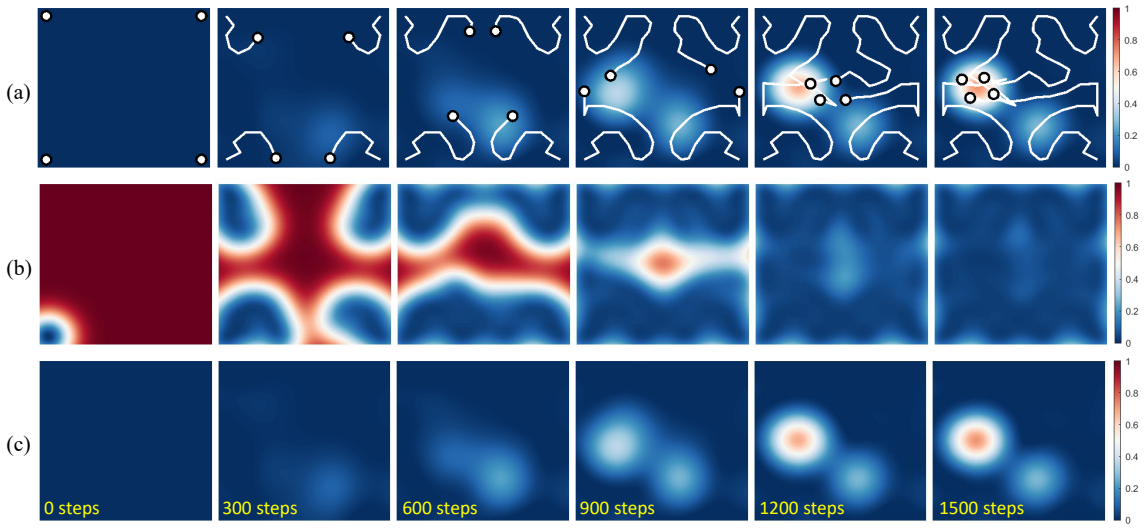


Fig. 4: Robotic sensor networks simulation for four agents to search for the highest peak location. It shows (a) the environmental process estimate and the trajectories of the agents, (b) the uncertainty propagation, and (c) the change of the GP estimate with time from $k = 0$ to $k = 1500$ in order from the left figure. The environmental model is presented in Fig 3. All results are estimates of agent $\#1$ initially located at bottom-left corner.

have a limit in the detailed description of the peak. To better describe the model around the peak, the exploitation step is needed [22]. Considering the confidence region of GP, exploration and exploitation can be performed at the same time by adding the mean value estimate and variance estimate, such as (25) (lines 26-27 of Algorithm 2).

$$\theta_{d,i} = \underset{\theta_c \in \Theta_c}{\arg\max} (\hat{f}_{E,i}(l \angle \theta_c + \mathbf{x}_i) + \eta \Sigma_{E,i}(l \angle \theta_c + \mathbf{x}_i)) \tag{25}$$

$\eta \in \mathbb{R}_{\geq 0}$ defines the confidence interval of GP model.

Local maxima can still happen in a rare situation when all the following conditions hold : i) the exploration range of individual agents, $l_{\text{threshold}}$ (in Algorithm 2), is very small, so the distance between all agents and a global maximum is greater than $l_{\text{threshold}}$ ($l_{\text{threshold}}$ can be limited by computational resource), ii) the variance of global maximum is small enough, so the agent can no longer access it, and iii) the estimation result of a global maximum by GP is incorrectly lower than local maxima.

B. Collision Avoidance and Coordination

For safe multi-robot exploration, we calculate the collision avoidance condition and the coordination weight. Robot i is assumed to know the locations of neighbor robots $j \in \mathcal{N}_i$ within the communication range l_{comm}.

In (24), the region farther than the collision-free distance l_{ca} is set as collision-free area Θ_{ca} (lines 7-8 of Algorithm 2). The workspace boundary condition is also applied in the same way (lines 5-6 of Algorithm 2).

For efficient coordination of multiple robots during the exploration, each agent gives the weights to the directions as far away from its neighbors as possible (lines 23-24 of Algorithm 2). The weight function $w_i(\mathbf{x_c})$ for agent i is defined as follows:

$$w_i(\mathbf{x}_c) = \sum_{j \in \mathcal{N}_i} (||\mathbf{x}_j - \mathbf{x}_c||)/(||\mathbf{x}_i - \mathbf{x}_c||) \tag{26}$$

It gives a linear weight with respect to $||\mathbf{x}_j - \mathbf{x}_c||$, which is scaled to 1 when the ratio of $||\mathbf{x}_j - \mathbf{x}_c||$ to $||\mathbf{x}_i - \mathbf{x}_c||$ is 1.

V. SIMULATION RESULT

This section presents a simulation on the virtual environmental model, as shown in Fig. 3. This bi-modal environmental model is unknown a priori, and each robot should obtain the sensory data from the current robot location. Since the communication range l_{comm} is 1, some agents may not be able to communicate with each other. The goal of this simulation is to find the highest peak in the entire workspace.

A. Simulation: robotic sensor networks for 4 agents

In this simulation, we perform distributed GP and active sensing for four agents. They conduct exploration to obtain an estimate of the overall environmental map, considering collision avoidance and coordination, then complete the simulation after rendering the highest peak. We set $\sigma_s^2 = 1$ and $\Sigma = \text{diag}([0.02, 0.02])$ for the Gaussian kernel (7), and we set $E = 80$ for E-dimensional estimator (14) and (20).

Fig. 4 represents the progress over time from $k = 0$ to $k = 1500$. As shown in Fig. 4(a), four agents searched the map together and gathered at the peak position without colliding with one another. All agents moved as far as possible from neighbors to improve exploration efficiency. Also, they avoid being close to the places in which the estimate is already reliable. Fig. 4(b) shows the result of the online GP variance estimate for the agent $\#1$ starting from the left-bottom corner, with the result of uncertainty lowering in all regions over time. Similarly, Fig. 4(c) shows the GP mean value estimate of the agent $\#1$, which gradually converges to the estimation result similar to Fig. 3. The

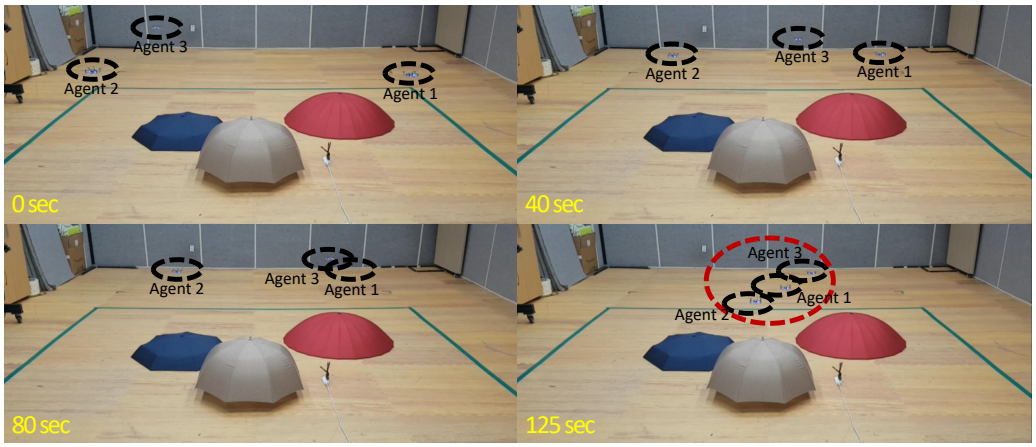


Fig. 5: 3 UAVs robotic sensor networks experiment to search the highest peak location. Snapshots from 0 to 125 seconds are shown. All UAVs complete the area exploration and at the end gather at the highest peak.

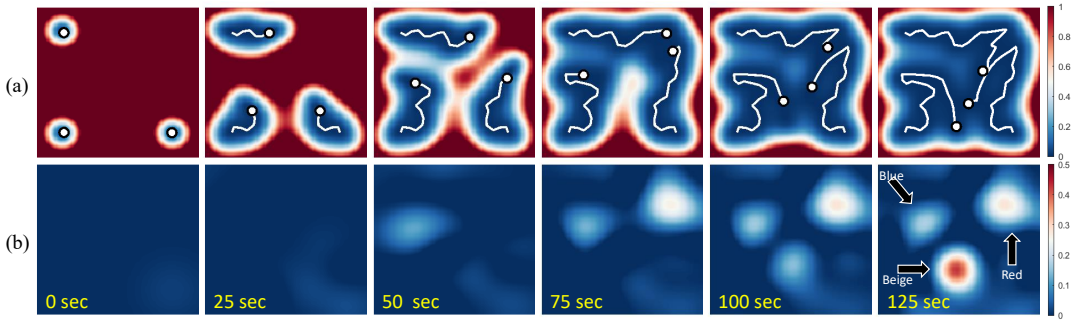
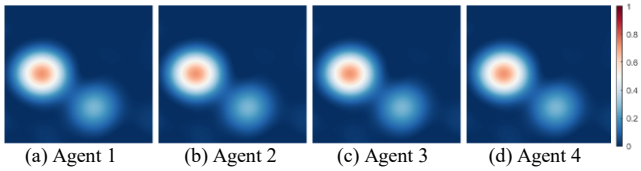


Fig. 6: Experiment: robotic sensor networks for three UAVs to search for the highest peak location. It shows (a) the uncertainty propagation and (b) the change of the GP estimate with time from 0 to 125 seconds in order from the left figure. All results are estimate of the agent #1.

exploitation process starts after $k = 1000$, and the shape of the peak becomes valid. By the average consensus, all the distributed GP estimates of each agent converge (see Fig.7).



(a) Agent 1 (b) Agent 2 (c) Agent 3 (d) Agent 4

Fig. 7: Final results of distributed GP for each agent. All GP mean estimates converge to the same because of the average consensus algorithm.

VI. EXPERIMENTAL RESULT

A. Experimental Setup

We validate our algorithm by experiment with multiple UAVs. Fig. 1 imitates the rough terrain to be used for the survey experiment. There are three hills with different height in a 4 m×4 m workspace marked with green lines, and the peak values are 39 cm (the beige), 29 cm (the red), and 19 cm (the blue), respectively. Three Crazyflie 2.1 nanocopters shown in Fig. 1 measure height above ground level (AGL) using VL53L1x Time-of-Flight sensor, which is mounted on the bottom of the Crazyflie. VL53L1x is a laser rangefinder that can measure a distance up to 4 meters, and the measurement error is from $\pm 1\%$ to $\pm 0.15\%$. The Time-of-Flight sensor noise is considered in GP calculation. VICON motion capture system and inertial navigation system (INS) in Crazyflie are used for indoor navigation. They fly at a constant altitude and perform topographic surveys over the entire workspace based on distance values measured with VL53L1x. The entire system is implemented in Robot Operating System (ROS), and the overall algorithm runs at 50 Hz including the online distributed GP. Because of the limited onboard computation resource of Crazyflie nanocopters, GP computation of all the individual UAVs is performed on a laptop (Intel i7-7500U CPU). The communication network is implemented based on the distance between agents, so such setting is not different from the distributed control scenario.

B. Experiment: robotic sensor networks for 3 agents

In this experiment, all UAVs perform an exploration to estimate the topographic map of the workspace, taking into account collision avoidance and coordination during exploration and complete the experiment at the highest peak (the center of the beige-colored umbrella). We set $\sigma_s^2 = 1$ and

$\Sigma = \text{diag}([0.02, 0.02])$ for the Gaussian kernel (7), and we set $E = 80$ for E-dimensional estimator (14) and (20).

Fig. 5 shows some snapshots taken during the experiment. All UAVs travel around the workspace, construct the topographic map using a laser rangefinder, and gather at the highest peak in the end.

Fig. 6 represents the progress of this experiment during $t = 125$ seconds. Fig. 6(a) shows the result of the online GP variance estimate for the agent $\#1$, with the result of uncertainty lowering in all regions over time except that the boundaries still had high uncertainty because of flight zone restriction of UAVs. Similarly, Fig. 6(b) shows that the GP mean value estimate of the agent $\#1$ gradually converges to the estimation result similar to Fig. 1. The exploitation process starts after about 100 seconds, and the shape of the beige region becomes sharp in Fig. 6(b).

VII. CONCLUSIONS

In this paper, we present a distributed GP and multi-robot sensor networks to obtain a global environmental model estimate. With the kernel expansion and average consensus algorithm, we implement online distributed GP in a networked control system. Using GP estimate, we establish a multi-agent exploration and exploitation algorithm with collision avoidance and coordination. Then, multi-robot exploration simulation is performed in a virtual environment, and an experiment is performed to construct a topographic map with multiple UAVs. This study is limited to a stationary environmental model, and we are working on expanding this work into a dynamic environment as future work.

APPENDIX
PROOF OF THEOREM 1

Proof: The distributed GP estimator (17) has to meet the following condition:

$$\frac{G^T G}{Nm} = \frac{1}{N} \sum_{i=1}^{N} \alpha_i(k) \tag{27}$$

for $k \geq k_0$. When all agents obtain m-th measurement $\{y_i(k_{m-1})\}_{i=1}^N$ at k_{m-1}, the left term in (27) can be expressed as follows:

$$\begin{aligned} \frac{G^T G}{Nm} &= \frac{1}{Nm} \sum_{i=1}^{N} \sum_{j=0}^{m-1} \Phi(\mathbf{x}_i(k_j)) \Phi^T(\mathbf{x}_i(k_j)) \\ &= \frac{1}{N} \sum_{i=1}^{N} \big(\tfrac{m-1}{m} \alpha_i(k-1) \\ &\quad + \tfrac{1}{m} \Phi(\mathbf{x}_i(k_{m-1})) \Phi^T(\mathbf{x}_i(k_{m-1})) \big) \\ &= \frac{1}{N} \sum_{i=1}^{N} \alpha_i(k), \quad \text{(using (23))} \end{aligned} \tag{28}$$

for $k_m > k \geq k_{m-1}$ and $m \geq 1$. Thus, (27) is satisfied at all times $k \geq k_0$. With the help of average consensus algorithm, $\{\alpha_i(k)\}_{i=1}^N$ move towards their average value, i.e., $\lim_{k\to\infty} \alpha_i(k) = \sum_{i=1}^N \alpha_i(k)$. As a result, $\{\alpha_i(k)\}_{i=1}^N$ becomes equal to the leftmost term in (22a). The update rule for $\beta_i(k)$ is the same. ■

REFERENCES

[1] Pajares and Gonzalo. "Overview and current status of remote sensing applications based on unmanned aerial vehicles (UAVs)." in *Photogram. Eng. & Remote Sensing*, 81.4 (2015): 281-330.

[2] B. J. Julian, M. Angermann, M. Schwager, and D. Rus. "Distributed robotic sensor networks: An information-theoretic approach." *Int. J. Robot. Res.*, 31.10 (2012): 1134-1154.

[3] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry. "Kalman filtering with intermittent observations." *IEEE tran. Auto. Cont.*, 49(9), (2004): 1453-1464.

[4] V. Gupta, B. Hassibi, and R. M. Murray. "On sensor fusion in the presence of packet-dropping communication channels." *Proc. 44th IEEE Conf. Dec. and Cont.*, 2005.

[5] A. Jadbabaie, J. Lin, and A. S. Morse. "Coordination of groups of mobile autonomous agents using nearest neighbor rules." *IEEE Trans. on auto. cont.* 48.6 (2003): 988-1001.

[6] R. O. Saber, and R. M. Murray. "Consensus protocols for networks of dynamic agents." (2003): 951-956.

[7] C. E. Rasmussen. "Gaussian processes in machine learning." *Summer Sch. Mach. Learn.*, Springer, 2003.

[8] M. P. Deisenroth, D. Fox, and C. E. Rasmussen. "Gaussian processes for data-efficient learning in robotics and control." *IEEE trans. patt. analysis and mach. intell.*, 37.2 (2013): 408-423.

[9] N. Chen, Z. Qian, I. T. Nabney, and X. Meng, "Wind power forecasts using Gaussian processes and numerical weather prediction." *IEEE Trans. Power Sys.*, 29.2 (2013): 656-665.

[10] H. Bijl, J. W. van Wingerden, T. B. Schön, and M. Verhaegen, "Online sparse Gaussian process regression using FITC and PITC approximations." *IFAC-PapersOnLine*, 48(28), (2015): 703-708.

[11] D. Nguyen-Tuong, J. R. Peters, and M. Seeger. "Local gaussian process regression for real time online model learning." in *Proc. Advances Neural Inf. Process. Syst.*, 2009, pp. 1193-1200.

[12] Z. Jin, and R. M. Murray. "Multi-hop relay protocols for fast consensus seeking." *Proc. 45th IEEE Conf. Dec. and Cont.*, 2006.

[13] S. Manfredi, "Design of a multi-hop dynamic consensus algorithm over wireless sensor networks." *Control Engineering Practice*, 21.4 (2013): 381-394.

[14] G. Pillonetto, L. Schenato, and D. Varagnolo. "Distributed multi-agent Gaussian regression via finite-dimensional approximations." *IEEE trans. patt. analysis and mach. intell.*, 41.9 (2018): 2098-2111.

[15] K. C. Ma, L. Liu, and G. S. Sukhatme. "Informative planning and online learning with sparse gaussian processes." *IEEE Int. Conf. Robot. and Auto. (ICRA)*, 2017.

[16] W. Luo, C. Nam, G. Kantor, and K. Sycara. "Distributed Environmental Modeling and Adaptive Sampling for Multi-Robot Sensor Coverage." *Proc. 18th Int. Conf. Auto. Ag. and MultiAg. Sys.*, 2019.

[17] A. Viseras, T. Wiedemann, C. Manss, L. Magel, J. Mueller, D. Shutin, and L. Merino. "Decentralized multi-agent exploration with online-learning of gaussian processes." *IEEE Int. Conf. Robot. and Auto. (ICRA)*, 2016.

[18] J. Choi, S. Oh, and R. Horowitz. "Distributed learning and cooperative control for multi-agent systems." *Automatica* 45.12 (2009): 2802-2814.

[19] Z. Huaiyu, W. Christopher K. I., R. Richard, and M. Michal. "Gaussian regression and optimal finite dimensional linear models." *Technical Report.* Aston University, Birmingham, 1997.

[20] R. Allamraju and G. Chowdhary. "Communication efficient decentralized Gaussian process fusion for multi-UAS path planning." *American Cont. Conf. (ACC)*, 2017, pp. 4442-4447.

[21] B.C. Levy. "Karhunen Loeve Expansion of Gaussian Processes." *Principles of Signal Detection and Parameter Estimation*. Springer, Boston, MA, 2008: 1-47.

[22] S. Ishii, W. Yoshida, and J. Yoshimoto. "Control of exploitation–exploration meta-parameter in reinforcement learning." *Neural networks* 15.4-6 (2002): 665-687.

[23] G. Flaspohler, V. Preston, A. P. M. Michel, Y. Girdhar, and N. Roy. "Information-Guided Robotic Maximum Seek-and-Sample in Partially Observable Continuous Environments." *IEEE Robot. and Auto. Letters* 4.4 (2019): 3782-3789.