

Remote Estimation for Dynamic IoT Sources under Sublinear Communication Costs

Jihyeon Yun, Atilla Eryilmaz, Jun Moon, and Changhee Joo

Abstract—We investigate a remote estimation system with communication cost for multiple Internet-of-Things sensors, in which the state of each sensor changes according to a Wiener process. Under sublinear communication cost structure, in which the per-transmission cost decreases with the number of simultaneous transmissions, we address an interesting unexplored trade-off under source dynamics between frequent updates of a smaller number of sensors at a higher cost and sporadic updates of a larger number of sensors at a lower cost. We first suggest two benchmark strategies, an all-at-once policy and a multi-threshold policy, and generalize them to a unified framework, called the MAX- k policy. Furthermore, we address the problem of parameter optimization of the MAX- k policy by developing online learning algorithms with stochastic feedback and a continuous search space. Through simulations, we demonstrate that the joint solution of the MAX- k policy and particle swarm optimization-based online learning achieves a high performance, outperforming the well-known upper confidence bound-based competitor.

Index Terms—Remote sensing, communication system control, Internet of Things

I. INTRODUCTION

Future Internet-of-Things (IoT) networks will consist of a large number of devices with internal sensors. Sensor information evolves over time and needs to be tracked at a remote location. Accordingly, remote estimation for the freshness of information for IoT applications is essential in many domains of IoT networks [2], including the following.

◇ *Mobile Healthcare Services*: As the functionality and capabilities of sensory devices (e.g., Apple watch, Fitbit, etc.) improve, they can monitor many different biological metrics, such as heart-rate, blood pressure, and body temperature, that must be transferred to a remote location over wireless channels to track the health state of the user.

Jihyeon Yun and Changhee Joo are with the Department of Computer Science and Engineering, Korea University, South Korea e-mail: (jihyeonyun@korea.ac.kr; changhee@korea.ac.kr)

Atilla Eryilmaz is with the Department of Electrical and Computer Engineering, The Ohio State University, USA e-mail: (eryilmaz.2@osu.edu)

Jun Moon is with the Department of Electrical Engineering, Hanyang University, South Korea e-mail: (junmoon@hanyang.ac.kr)

This work was supported in part by the NRF grants funded by the Korea government (MSIT) (No. NRF-2021R1A2C2013065 and 2022R1A5A1027646) and by the MSIT, Korea, under ICT Creative Consilience program (IITP-2021-2020-0-01819) supervised by IITP. Also, it was supported in part, by NSF grants of NSF AI Institute (AI-EDGE) 2112471, CNS-NeTS-2106679, CNS-NeTS-2007231, and by the ONR Grant N00014-19-1-2621.

A preliminary version of this work appeared in IEEE INFOCOM'21 Age of Information Workshop [1]

Changhee Joo is the corresponding author.

◇ *Intelligent Transportation Networks*: Vehicles in a transportation network have locally evolving states, such as position, direction, and proximity to other objects around it, that are envisioned to be tracked closely by the infrastructure network (such as a roadside unit) over a wireless communication medium.

◇ *Internet of Military Things (IoMT)*: The military devices have sensors to detect and measure information for surroundings, such as auditory, visual, and heat, and generate fresh updates of new information to coordinate and interact with physical environment to achieve military activities efficiently.

A common theme in all these scenarios is the necessity to efficiently transfer multiple evolving states from a transmitter to a receiver over a communication channel to closely track their states at the receiver while maintaining low communication costs. In this paper, we discuss this generic problem in the key case of N independent Wiener processes describing the source dynamics, which is reasonable because the Wiener process is used to represent the integral of the Gaussian white noise process (e.g., gyroscope drift [3]); therefore, it is useful to explain noise and errors in many IoT systems. We also assume that each evolving state follows an independent Wiener process, where each sensor information is collected from different sources, and a center transmitter aggregates them and sends the information to a remote receiver. For example, the healthcare monitoring system that has a wireless access point that serves as a cluster head and multiple sensors that collect the status of patients would be a good example. In our work, we target such a remote estimation system and try to optimize the average system cost.

In related works in this domain, remote estimation under asynchronous massive access of IoT sensors in a mission-critical manner was studied in [4], and state estimation for IoT-based vehicles under cyber-attacks was considered in [5]. Furthermore, a remote estimation system that considers the average mean square error (MSE) over a stochastic process was analyzed with different constraints in [6]–[15]. A joint problem of scheduling and remote estimation that minimizes communication costs over a finite time horizon was formulated in [6]. The problem was extended to an energy-harvesting sensor in [7], and with constraints on the number of transmissions in [8]. Furthermore, in [9], a noise channel was considered, with and without communication costs. Similarly, the remote estimation problem of a stochastic process with one perfect (but costly) communication channel and one noisy (but cheap) communication channel was considered in [10]. The authors found that the optimal policy was a threshold-in-threshold scheduling policy with some assumptions. For continuing (in-

finite time horizon) tasks, the studies of [11], [12] investigated the cost minimization problem in a remote estimation system with a packet drop channel and communication cost, respectively. More recently, in [13], [14], the MSE was minimized for a one-dimensional Wiener process under a sampling frequency constraint over an infinite-time horizon. In [15], a cost minimization problem with multi-dimensional Wiener processes under constraints on communication frequency was addressed. Remote estimation problems for systems with a collision channel (i.e., only one sensor state can be successfully transmitted through the channel) had been investigated in [16]–[18]. In these problems, multiple sensors tried to send their sensor states to the estimator through the channel and sampled their sensor values while minimizing the probability of estimation error under collision [16], [17]. Further, the authors of [18] designed threshold strategies for a remote estimation system with collision channel and analytically found a unique optimal threshold under some assumptions. A distributed learning was also exploited in [18] to find the optimal thresholds when partial distribution knowledge for sensor states was given. In contrast to the above works, we consider a perfect channel and focus on the trade-off between the estimation error and the communication cost under sublinear communication costs. In the field of information theory and control, the trade off between communication rate (bit/s) and the expected system cost was investigated in [19] considering linear quadratic regulator (LQR), where the authors provided the lower bound on the rate-cost function which they proposed. Also, the trade-off between the system cost and the communication resources was studied in [20] for a linear quadratic Gaussian (LQG) problem with an additional communication channel between encoder and decoder. The optimization problem to minimize the directed communication under a constraint on the system cost was investigated and it was shown that the problem has a standard convex form. Further, the authors of [21] considered a system model where an encoder samples continuous Markov Processes and transmits codewords to a decoder. In [21], the optimal encoding policy to minimize the estimation error under a constraint of the communication rate was proven to be a threshold-based policy.

In this paper, we consider a remote estimation system over N independent Wiener processes, which can be considered an N -dimensional Wiener process, with a communication cost. If the communication cost is proportional to the number of processes to be transmitted simultaneously, the multi-dimensional nature of the problem can be simply decomposed into multiple remote estimation problems of tracking a single Wiener process. However, if the communication cost has a sublinear form in the setting outlined above, i.e., if the cost for transmitting $k+1$ units of information is less than the cost for transmitting k units plus the cost for 1 unit (see Section II), the problem cannot be simply decomposed. In this paper, we focus on remote estimation with sublinear communication costs for two reasons. First, the wireless transmission of multiple information units often provides gains in per-unit power consumption, and wireless IoT devices commonly have a limited power budget. Second, most IoT networks consist of a large number of devices compared with limited network

resources. Thus, collisions due to simultaneous transmissions are likely to occur more frequently, which deteriorates the utilization of already-limited resource [22]. Transmitting multiple information units can improve the resource efficiency by reducing the chance of collision.

We consider a remote estimation system in which multi-dimensional Wiener process (or multiple independent Wiener processes) is estimated under a sublinear communication cost function. The sublinear form of the communication cost function provides a new trade-off between the estimation error and communication cost. Frequently updating a small subset of information can effectively reduce the estimation error; however, it incurs a higher communication cost per unit. However, sporadically updating a larger subset of information requires a lower communication cost per unit, but it is likely to have a higher estimation error. Therefore, we first consider two extreme-case benchmark strategies: an all-at-once updating policy and a multi-threshold policy that conducts one update at a time. We generalize them and develop a novel strategy called the MAX- k policy, which outperforms the two benchmarks if configured accordingly. However, setting the optimal parameters of the MAX- k policy requires prior information regarding the estimation error and communication cost function, which are commonly unavailable in practice.

This motivated us to develop an online stochastic learning algorithm that can be combined with the MAX- k policy. We exploit the well-known upper confidence bound (UCB) algorithm and the particle swarm optimization (PSO) technique and extend them to account for stochastic outcomes in a continuous search space. We observe that PSO-based learning converges significantly faster than UCB-based learning. Finally, we show the performance of the joint solution of the MAX- k policy and online learning, and we numerically study the impact of a large N and different degrees of sublinearity of the communication cost function.

The preliminary version of this work has been presented in the workshop [1]. We substantially extends it with the convergence of the proposed scheme, detailed explanations and performance analysis of comparable policies, and extended simulations for the performance evaluation. The remainder of this paper is organized as follows. The system model is described in detail in Section II. In Section III, two benchmark policies are introduced and the MAX- k policy is developed with a performance comparison. In Section IV, online stochastic learning algorithms for optimizing the parameters of MAX- k policy are provided with provable convergence. In Section V, the performance of the MAX- k policy with online algorithms is evaluated through simulations. We conclude the paper in Section VI.

II. SYSTEM MODEL

We consider a remote estimation system with a transmitter, a remote receiver, and N sensors (Fig. 1). The sensors have randomly evolving states and each state value follows an i.i.d Wiener process. The transmitter can observe and collect sensor state values and transmit the collected information to the receiver in an aggregated frame through a communication

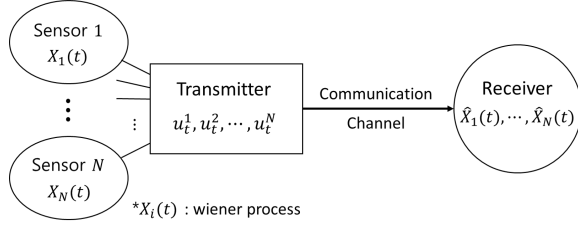


Fig. 1. Remote estimation system.

channel. Communication costs are explained below. The remote receiver receives the information and tracks the state of the sensors to make a certain decision in a timely manner. A more detailed description of our system is provided in the remainder of this section.

The transmitter is responsible for each transmission. At each time, the transmitter decides which sensor values to be sent to the receiver or decides not to transmit at all. The receiver tracks N sensor values by updating each sensor value with the most recently received value from the transmitter.¹ The process of transmitting and updating sensor values incurs two different types of system cost. One is the communication cost, and the other is the estimation cost owing to the staleness of the estimated sensor values at the receiver. We assume that the communication cost is a sublinear function of the number of collected sensor values in a transmission. Some examples are discussed later in this paper. For the estimation cost, we consider the widely used MSE between the true state values at the sensors and the estimated sensor values at the receiver. We aim to minimize the total sum of communication and estimation costs. Various performance metrics can be used for the system cost depending on the models and the applications. For example, Age of Information can be included as a system cost if the system has high sensitivity to the information freshness. We can also consider collision rate and throughput for evaluating the system performance when the communication channel is shared with other networks.

We now provide a formal description of this problem in the following. We also summarize some terms that are frequently used in this paper in Table I. Let $X_i(t)$ be the true state value of sensor i at time t , which evolves as a standard Wiener process in continuous time, independently across sensors, i.e.,

$$X_i(t + \alpha) - X_i(t) \sim \mathcal{N}(0, \alpha),$$

for all $t \geq 0$ and $\alpha \geq 0$, where $\mathcal{N}(\mu, \sigma)$ is a normal distribution with a mean μ and variance σ . We assume $X_i(0) = 0$ for all i .

At each time t , the transmitter transmits a single frame with n_t number of sensor values. The inclusion of sensor i 's value is denoted by the binary variable u_t^i . Specifically, $u_t^i = 1$ means that the sensor value $X_i(t)$ is transmitted to the receiver, and $u_t^i = 0$ means that it is not. We have $n_t = \sum_{i=1}^N u_t^i$. Let $\hat{X}_i(t)$ denote the estimated sensor value for sensor i at the

¹This is of practical use when the state changes smoothly according to an i.i.d normal distribution. For more complex state dynamics, the receiver may use an MAP (Maximum A Posteriori). See [23] for an example of an MAP receiver.

TABLE I
TERMINOLOGY.

Term	Used for
N	number of sensors
$X_i(t)$	true state value of sensor i
$\hat{X}_i(t)$	estimated state value of sensor i at the receiver
$\mathcal{E}_i(t)$	estimation error for sensor i
$\hat{\mathcal{E}}_i(t)$	predicted estimation error for sensor i
u_t^i	update decision for state value of sensor i
n_t	the number of transmitted state values
f	communication cost function
α	degree of sublinearity in f
γ^{all}	threshold under all-at-once policy
γ^{mul}	threshold under multi-threshold policy
$\gamma^{(k)}$	threshold under MAX- k policy
U	the number of transmissions that makes up a round
Δ	duration of a round
$\hat{r}$	average system cost obtained for a round
A	set of arms in MAX- k -UCB
C	set of cells in MAX- k -GPSO
h	function that maps a position to a cell
g	function that maps a cell to the expected average cost
$\bar{g}$	empirical mean of average cost over the cell
S	set of particles in a swarm
s^*	global best particle
$V(c)$	the number of visits of any particles to a cell c
$\mathbf{x}_s$	position of a particle s
$\mathbf{y}_s$	best position of particle s among its visits
$\hat{\mathbf{y}}$	global best position of all particles based on all visit history
$\mathbf{v}_s$	velocity of particle s

receiver, i.e., the most recently received $X_i(\tau)$ for $\tau \leq t$. We assume that the transmission error and transmission time are negligible. $\hat{X}_i(t)$ evolves as

$$\hat{X}_i(t) = \begin{cases} X_i(t), & \text{if } u_t^i = 1, \\ \hat{X}_i(t-1), & \text{if } u_t^i = 0. \end{cases} \quad (1)$$

We assume that $\hat{X}_i(0) = 0$ for all i . The estimation cost closely involves the information staleness or estimation error, which is defined as

$$\mathcal{E}_i(t) = X_i(t) - \hat{X}_i(t).$$

The estimation cost at time t is computed as the squared estimation error sum, i.e., $\sum_i \mathcal{E}_i^2(t)$. Frequent updating of the sensor values can reduce the estimation cost, which, implies frequent transmissions and results in higher communication costs.

We consider the communication cost function of the sub-linear structure in n_t , the number of transmitted sensor values. This is motivated by the fact that encoding more information into a transmission frame increases the coding rate, and by practical frame aggregation techniques: sending multiple frames simultaneously results in less transmission overhead [24] or a higher amount of saved energy [25]. It has been known that a transmission of a single packet involves many overheads from multiple places, which include additional header structure added at each protocol stack, contention resolution for resource sharing, and the saturation time of RF module for signal transmission. For example, in Wi-Fi systems, we can save at least 10% energy or improve throughput by three times by aggregating multiple packets into

one [26], [27]. Specifically, we model the communication cost using the following concave function:

$$f(n_t) = \bar{c} \cdot (n_t)^\alpha, \quad (2)$$

where $\bar{c} > 0$ is a constant and $0 < \alpha \leq 1$ is an exponent that represents the level of sublinearity. Note that the communication cost becomes 0 when the transmitter does not transmit any sensor values (i.e., $n_t = 0$). Our aim is to minimize the expected total average cost R defined as

$$R = \lim_{T \rightarrow \infty} \mathbb{E} \left[\frac{1}{T} \int_0^T \left(\sum_{i=1}^N \mathcal{E}_i^2(t) + f(n_t) \right) dt \right]. \quad (3)$$

Several studies have addressed similar problems. For one-dimensional stochastic process ($N = 1$), studies have shown that a threshold policy is an optimal solution [6]–[14]. For the multi-dimensional (or multiple independent processes) case, when all sensor values should be transmitted together, a threshold-based update policy is optimal [15]. Our problem is different from these studies in that individual sensors may transmit their values, and the communication cost is a sublinear function of the number of sensor values being transmitted. This unexplored shape of communication costs is vital in practice, and motivates interesting new policy designs. For the following sections, we investigate threshold-based policies for our remote estimation problem under sublinear communication cost function. Inspired by the existing studies [6]–[15] that showed a threshold policy is an optimal solution, we conjecture that a threshold-based policy performs well under sublinear communication cost and develop a novel MAX- k policy to improve the performance.

III. THRESHOLD-BASED POLICIES

We first introduce two benchmark policies, and then develop a generalized policy that can incorporate them as an extreme case. We show that the two benchmark policies are extreme cases, and that none of them can be an optimal policy under the sublinear communication cost function (2). To facilitate this explanation, we introduce $\tilde{\mathcal{E}}_i(t)$, which denotes a *predicted estimation error* at the transmitter. At each time t , before a decision $\{u_t^i\}_i$ is made for transmission, the transmitter predicts the estimation error as $\tilde{\mathcal{E}}_i(t)$ assuming that it does not send the state value of sensor i at time t , which can be expressed as

$$\tilde{\mathcal{E}}_i(t) = X_i(t) - X_i(t'),$$

where t' be the time of the latest update for sensor i .

A. All-at-once policy

First, we define the *all-at-once policy* that first appeared in [15]. Under the all-at-once policy, the number of transmitted sensor values n_t should be either 0 (no transmission) or N (transmission of all sensor values) at each time t . The decision variables of the transmitter can then be represented by one variable u_t as $u_t = u_t^1 = u_t^2 = \dots = u_t^N$. It has been shown

in [15] that an optimal policy under the constraint $n_t \in \{0, N\}$ is of the threshold type:

$$u_t = \begin{cases} 1, & \text{if } \sqrt{\sum_i \tilde{\mathcal{E}}_i(t)^2} \geq \gamma^{all}, \\ 0, & \text{if } \sqrt{\sum_i \tilde{\mathcal{E}}_i(t)^2} < \gamma^{all}, \end{cases}$$

with threshold $\gamma^{all} = \sqrt[4]{2(N+2)f(N)}$. We can also obtain the optimal average cost R^{all} as

$$R^{all} = \sqrt{\frac{2N^2 f(N)}{N+2}}. \quad (4)$$

B. Multi-threshold policy

Another natural extreme policy is an independent decision maker for each sensor. When the state value of sensor i changes, a predicted estimation error $\tilde{\mathcal{E}}_i(t)$ is calculated and the transmission decision for sensor i is expressed as

$$u_t^i = \begin{cases} 1, & \text{if } |\tilde{\mathcal{E}}_i(t)| \geq \gamma_i^{mul}, \\ 0, & \text{if } |\tilde{\mathcal{E}}_i(t)| < \gamma_i^{mul}. \end{cases} \quad (5)$$

For identical sensors, we have the same threshold value for all i (i.e., $\gamma_i^{mul} = \gamma^{mul}$ for all i) because the state value of each sensor follows an independent standard Wiener process. For this single-sensor system, the system behavior was well studied and the threshold-type policy (5) was shown to be the optimal solution in [6]. However, this multi-threshold policy fails to achieve the optimal performance in our problem because of the sublinear communication cost.

Let us estimate the average cost under a multi-threshold policy. Although the communication cost of transmitting n sensor values simultaneously is smaller than n times the cost of transmitting 1 sensor value, we can replace $f(n_t)$ with $\sum_{i=1}^N f(u_t^i)$ in Eq.(3) because the continuous nature and the independence of the Wiener processes admit no simultaneous update. Thus, we can rephrase Eq.(3) as the following.

$$\lim_{T \rightarrow \infty} \mathbb{E} \left[\frac{1}{T} \int_{t=0}^T (\mathcal{E}_i^2(t) + f(u_t^i)) dt \right],$$

for each $i \in \{1, \dots, N\}$. We drop the subscript i for brevity. Let τ be the first time when $|\tilde{\mathcal{E}}(t)|$ hits the threshold γ from $\mathcal{E}(t) = 0$. The estimation error $\mathcal{E}(t)$ can be considered a renewal process whose renewal intervals restart with each update. The average cost can be obtained by calculating the expectation during the first renewal interval, i.e.,

$$\mathbb{E} \left[\frac{1}{\tau} \left(\int_{t=0}^{\tau} \mathcal{E}(t)^2 dt + f(1) \right) \right].$$

From the properties of renewal processes [28], this can be expressed as

$$\frac{\mathbb{E}[\int_0^{\tau} \mathcal{E}(t)^2 dt]}{\mathbb{E}[\tau]} + \frac{f(1)}{\mathbb{E}[\tau]}. \quad (6)$$

We first calculate $\mathbb{E}[\int_0^{\tau} \mathcal{E}(t)^2 dt]$. Using the Itô lemma [29], for any t , we obtain

$$d(t\mathcal{E}(t)^2) = \mathcal{E}(t)^2 dt + 2t\mathcal{E}(t)d\mathcal{E}(t) + tdt.$$

Integrating it up to τ and rearranging the equation, we obtain

$$\int_0^{\tau} \mathcal{E}(s)^2 ds = \tau\mathcal{E}(\tau)^2 - \frac{1}{2}\tau^2 - \int_0^{\tau} 2s\mathcal{E}(s)d\mathcal{E}(s).$$

We take the expectations of both sides. The last term on the right side then disappears because $\mathbb{E}[\int_0^\tau 2s\mathcal{E}(s)d\mathcal{E}(s)] = 0$ [29]. Thus, we obtain

$$\mathbb{E}[\tau\mathcal{E}(\tau)^2 - \frac{1}{2}\tau^2] = \mathbb{E}[\int_0^\tau \mathcal{E}(s)^2 ds] = \frac{1}{6}\mathbb{E}[\mathcal{E}(\tau)^4],$$

where the last equality was obtained from [13]. Because $\mathcal{E}(\tau) = \gamma$, we have $\mathbb{E}[\int_{t=0}^\tau \mathcal{E}(t)^2 dt] = \frac{1}{6}\gamma^4$. Subsequently, using $\mathbb{E}[\tau] = \gamma^2$ [30], we can express (6) as $\frac{\gamma^2}{6} + \frac{f(1)}{\gamma^2}$, which is minimized with the optimal threshold $\gamma^{mul} = \sqrt[4]{6f(1)}$. Hence, we can obtain the optimal average cost R^{mul} as

$$R^{mul} = \frac{\sqrt{6}}{3} \sum_{i=1}^N \sqrt{f(1)}. \quad (7)$$

From (4) and (7), we can observe that the all-at-once policy achieves a lower cost than the multi-threshold policy when $\alpha < \log_N \frac{N+2}{3}$. This implies that when α is sufficiently small, the gain from simultaneous updates overwhelms the loss owing to an inaccurate estimation. In the next section, we further investigate the trade-off between communication cost and estimation cost by developing a novel policy that generalizes the two benchmark policies.

C. Generalized threshold-based policy: MAX- k policy

Each of the aforementioned two benchmark policies are an extreme case – transmitting all N sensor values together, or transmitting one sensor value at a time. We develop a new transmission policy, MAX- k policy, that generalizes two benchmark policies and allow $k \in [1, N]$ sensor values at a time. Under MAX- k policy, the transmitter can transmit k sensor values with k highest expected estimation errors at a time. Each transmission is also determined on the basis of a threshold. A formal description of MAX- k policy is as follows.

Let $\{\pi_t^i\}_{i=1}^N$ denote the permutation of the sensors at time t in the order of $\tilde{\mathcal{E}}_i(t)$, satisfying $\tilde{\mathcal{E}}_{\pi_t^i} \geq \tilde{\mathcal{E}}_{\pi_t^{i+1}}$ where a tie can be broken arbitrarily. Furthermore, we define $A_k(t)$ as the set of the largest k elements of the permutation, i.e., $A_k(t) = \{\pi_t^1, \pi_t^2, \dots, \pi_t^k\}$.

Definition 1. MAX- k policy is a threshold-based policy that updates state values of k out of N sensors as

$$u_t^i = \begin{cases} 1, & \text{if } \sqrt{\sum_{j \in A_k(t)} \tilde{\mathcal{E}}_j(t)^2} \geq \gamma \text{ and } i \in A_k(t), \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

When the MAX- k policy achieves the minimum cost (3), we denote the associated threshold as the optimal threshold $\gamma^{(k)}$.

The notion of optimality in Definition 1 is within the class of MAX- k type policies. Note that the MAX- k policy is equivalent to the multi-threshold policy when $k = 1$ and to the all-at-once policy when $k = N$. Consequently, the optimal threshold of the MAX- k policy should satisfy $\gamma^{(1)} = \gamma^{mul}$ and $\gamma^{(N)} = \gamma^{all}$. The first question is whether this generalization improves performance in terms of the expected average system cost in comparison with the two extreme policies, and in what conditions. Another interesting question is how to determine the optimal values of parameters k and $\gamma^{(k)}$, if we have no prior knowledge of the sensor dynamics and communication cost function. In the remainder of this section, we answer the

first question by numerically evaluating the performance of the MAX- k policy with different α values of the communication cost function in (2). We observe that the MAX- k policy with a proper $k \in \{2, \dots, N-1\}$ and $\gamma^{(k)}$ significantly outperforms the benchmark policies in a certain interval of α . Note that we consider constant threshold $\gamma^{(k)}$ for MAX- k policy, assuming static network environment. If the network environment changes dynamically or the objective includes a hard constraint, the optimal threshold will change accordingly. In this work, we focus on static setting, and leave the problem of optimizing the threshold in dynamic environment as a future work.

We consider N sensors, where the state value of each sensor evolves independently and follows a discretized version of the standard Wiener process. The time is slotted, and the state values of the sensors have increments according to the normal distribution with mean 0 and variance 1 at each time slot, i.e.,

$$X_i(t) = X_i(t-1) + W_i(t) \text{ for } t \in \{1, 2, 3, \dots\},$$

where $W_i(t)$ is the increment at time t and $W_i(t) \sim \mathcal{N}(0, 1)$. We also assume $X_i(0) = 0$ for all i . At each time slot t , the transmitter makes an update decision u_t^i for each source i according to a certain policy. We use the communication cost function of (2) with $n_t = \sum_{i=1}^N u_t^i$ and $\bar{c} = 100$. Based on the transmission, the receiver estimates $X_i(t)$ with $\hat{X}_i(t)$ according to (1). Using the estimation error $\mathcal{E}_i(t) = X_i(t) - \hat{X}_i(t)$, the total cost at time t is $\sum_{i=1}^N \mathcal{E}_i^2(t) + f(n_t)$ and the average cost from $t = 0$ to $t = T$ is

$$\frac{1}{T+1} \sum_{t=0}^T \left(\sum_{i=1}^N \mathcal{E}_i^2(t) + f(n_t) \right), \quad (9)$$

where $\mathcal{E}_i(0) = 0$ for all i and $f(n_0) = 0$. For all-at-once and multi-threshold policies, we use their optimal threshold levels γ^{all} and γ^{mul} , respectively. For the MAX- k policy, we assume that γ is an integer and empirically determine the best-performing γ for each $k \in [N]$ where $[N] := \{1, 2, \dots, N\}$. Let $\tilde{\gamma}^{(k)}$ denote the empirical integer threshold value that achieves the lowest average cost under MAX- k policy. We used $\tilde{\gamma}^{(k)}$ only for the preliminary simulations and we will later introduce a method to determine $\gamma^{(k)}$ without an integer assumption. For each setting, we repeated 30 simulation instances, each of which continued for 5000 time slots, and obtained the empirical mean of the average cost (9).

Fig. 2 shows the best-performing integer threshold for the MAX- k policy, $\tilde{\gamma}^{(k)}$, for each $k \in [N]$ with $N = 10$, and different α values. The value of $\tilde{\gamma}^{(k)}$ does not decrease as k increases, which is reasonable because the optimal threshold will not decrease while the communication cost is an increasing function of the number of sensors. In addition, $\tilde{\gamma}^{(k)}$ does not decrease with respect to α , which is because an optimal policy will make less frequent transmissions (i.e., higher threshold) as the communication cost increases (i.e., α increases).

Fig. 3 shows the total average cost achieved by the MAX- k policy with different parameters of α and k when $N = 10$. For each k , the best-performing threshold $\tilde{\gamma}^{(k)}$ is used. For each α , we mark the best-performing k that provides the lowest average cost by a cross. Since MAX-1 and MAX-10 policies correspond to multi-threshold and all-at-once policies,

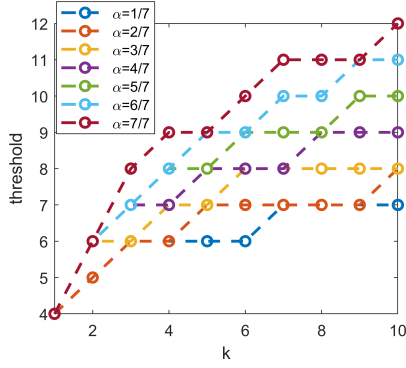


Fig. 2. Threshold for MAX- k policy with $N = 10$

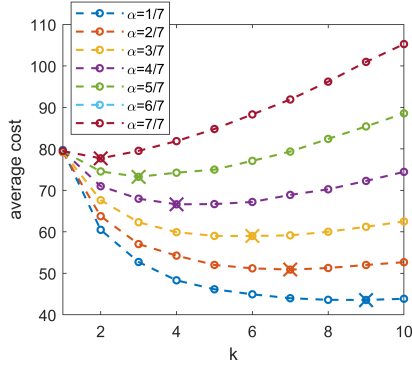


Fig. 3. Average cost of MAX- k policy with $N = 10$.

respectively, the left end points (when $k = 1$) show the average cost of the multi-threshold policy and the right end points (when $k = 10$) indicate the average cost of the all-at-once policy. We can observe that neither multi-threshold nor all-at-once policy is optimal. Furthermore, the best-performing k of the MAX- k policy changes for different α values and increases toward N as α decreases closer to 0. This means that for better performance, the number of sensors to be updated increases when the benefit of simultaneous updates increases (i.e., α decreases). If the same communication cost is charged regardless of the amount of transmitted information (i.e., when $\alpha = 0$), the all-at-once policy achieves the best performance.

Next, we compare the three policies for different N and α values. For the MAX- k policy, we use the best-performing k with the corresponding $\hat{\gamma}^{(k)}$. For the all-at-once and multi-threshold policies, we again use their optimal threshold levels γ^{all} and γ^{mul} , respectively. Fig. 4 shows the average cost of the three policies as the number of sensors increases (i.e., N increases). The average cost of the multi-threshold policy increases linearly with N , whereas the average cost of the all-at-once policy increases sublinearly. As expected, the MAX- k policy achieves the best performance in all cases. We note that the performance gap between the three policies increased as the number of sensors increases. In addition, as α increases, the performance gap between the MAX- k policy and the multi-threshold policy decreases, whereas the performance gap between the MAX- k policy and the all-at-once policy

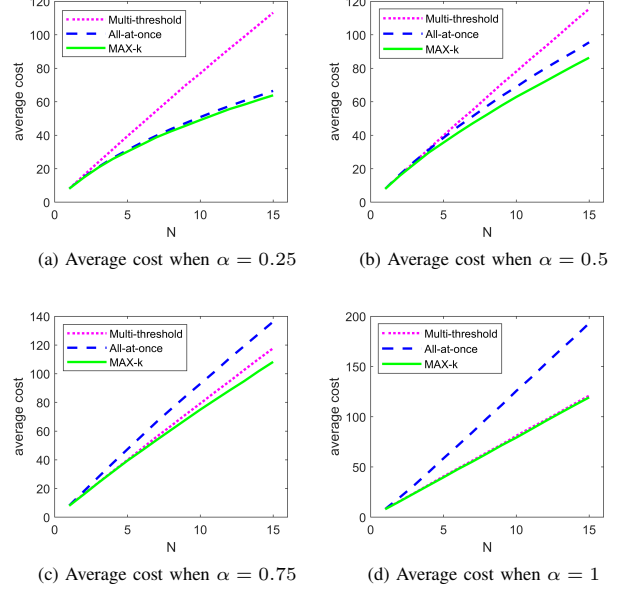


Fig. 4. Average cost of three policies.

increases. We notice that MAX- k policy is a sampling policy that is more efficient than multi-threshold and all-at-once policies under sublinear communication cost functions (i.e., $0 < \alpha < 1$). The performance gap between them becomes larger as we increase the number of sensors in the system.

IV. OPTIMIZING MAX- k POLICY

For optimal performance, determining the best-performing k and the corresponding optimal threshold level $\gamma^{(k)}$ in (8) is imperative for the MAX- k policy. Let k^* denote the best-performing k and $\gamma^{(k^*)}$ denote the corresponding optimal thresholds for the MAX- k policy. However, obtaining an analytical solution for k^* and $\gamma^{(k^*)}$ is difficult because of the complex and highly coupled dynamics of the system. Although the evolution of the sources is not coupled due to the independence of Wiener processes, their error dynamics is coupled with the decisions under MAX- k policy. The statistics of k highest-valued Wiener processes is needed to analyze the average system cost of MAX- k policy and calculate the optimal solution, k^* and $\gamma^{(k^*)}$, which is difficult despite the independence of the Wiener processes because the k highest-valued processes (among N processes) change over time. Furthermore, in many practical scenarios, we may have no information about the communication cost function or statistics of the evolving sensor states. Therefore, we use an online learning approach to determine $(k^*, \gamma^{(k^*)})$.

A straightforward method is to approximate the problem into a stochastic multi armed bandit (MAB) problem. If we fix the tuple (k, γ) and make U update decisions using the MAX- k policy with parameters k and γ , then the time for U updates, denoted by a *round*, and the average cost during the time define a stochastic process with a fixed distribution parameterized by (k, γ) . Hence, we can consider the tuple as an arm, and the resulting average cost during one round as its associated reward. The solution to this problem is to determine

the optimal arm $(k^*, \gamma^{(k^*)})$ that provides the minimum average cost under the MAX- k policy. In the following subsections, we develop two variants of MAX- k policies that optimize k and threshold values while proceeding with updates of sensor values: one is a policy that applies the UCB algorithm to the MAX- k policy, and the other applies a newly developed grid-PSO (GPSO) algorithm to MAX- k policy which will be shown to determine the optimal solution if the underlying objective function is convex.

A. Optimizing MAX- k policy with UCB index

A widely adopted approach to online learning of determining the optimal solution $(k^*, \gamma^{(k^*)})$ is to use the UCB index [31] for all possible arms (k, γ) . However, to apply the UCB algorithm, we face the problem of infinite arms owing to the real-valued threshold level γ . Note that we have a bounded real-valued range $\gamma^{(k)} \in [\gamma^{mul}, \gamma^{all}]$ as $\gamma^{(k)}$ increases as k increases, and $\gamma^{(1)} = \gamma^{mul}$ and $\gamma^{(N)} = \gamma^{all}$. We address this problem by partitioning the range into M intervals, denoted by $\mathcal{I} = \{I_m\}_{m=1}^M$ with $M < \infty$. The M intervals are exactly the same size, and each interval is centered at $\{\gamma_m\}_{m=1}^M$. In the following, we assume $\gamma \in \{\gamma_m\}_{m=1}^M$ unless otherwise stated. We extend the well-known UCB algorithm and apply it to our online learning for $(k^*, \gamma^{(k^*)})$ of the MAX- k policy, which is denoted by MAX- k -UCB.

We group consecutive U transmissions as a round, and consider the history of decisions and costs during a round as an episode. We assume that, at the end of each round, the transmitter sends all sensor values to the receiver to reset the estimation error for all sensors to 0, i.e., the communication cost at the end of a round is always $f(N)$ (not 0 or $f(k)$). This final update to reset the estimation error at each round is somewhat arbitrary, but its impact becomes negligible as U increases. For a sufficiently large U , we set $U = N^2$, and update each sensor value approximately N times during a round. Let Δ denote a random variable corresponding to the time duration of a round, and define $\hat{r}$ as the time-averaged system cost during a round, i.e.,

$$\hat{r} = \frac{1}{\Delta} \int_{\Delta} \left(\sum_{i=1}^N \mathcal{E}_i^2(t) + f(n_t) \right) dt. \quad (10)$$

Let A denote the set of available arms with a total of $|A|$ arms. Each arm a represents $(k, \gamma) \in R_1 \times R_2$ where $R_1 = [N]$ and $R_2 = \{\gamma_1, \dots, \gamma_M\}$. i.e., $|A| = N \times M$. Additionally, let $\bar{a}$ denote the selected arm during a round. Each arm a has three internal parameters: the index value $\text{UCB}(a)$, number of selection $\tau(a)$, and empirical average cost $\eta(a)$.

At the beginning of each round, MAX- k -UCB selects $\bar{a}$ as follows: For the first $|A|$ rounds, the policy selects each arm exactly once and then computes $\text{UCB}(a)$ for each a as

$$\text{UCB}(a) = \eta(a) - \sqrt{\frac{2 \log(\text{round})}{\tau(a)}},$$

where round denotes the number of rounds played. Thereafter, it selects $\bar{a} = \arg\min_a \text{UCB}(a)$. At the end of each round, it updates $\tau(\bar{a})$ and $\eta(\bar{a})$ accordingly. The process continues until a stopping condition is true (e.g., until time T is reached), and the arm with the smallest empirical average cost, which

Algorithm 1 MAX- k -UCB policy

Input: U , set of arms A

- 1: $\text{UCB}(a) = 0$, $\tau(a) = 0$, $\eta(a) = 0$ for all $a \in A$, and $\text{round} = 0$
 - 2: **while** stopping condition is not true **do**
 - 3: $\text{round} \leftarrow \text{round} + 1$
 - 4: **if** $\text{round} \leq |A|$ **then**
 - 5: $\bar{a} \leftarrow$ Pick an arm a with $\tau(a) = 0$ at random
 - 6: **else**
 - 7: $\text{UCB}(a) \leftarrow \eta(a) - \sqrt{\frac{2 \log(\text{round})}{\tau(a)}}$ for all $a \in A$
 - 8: $\bar{a} \leftarrow \arg\min_a \text{UCB}(a)$
 - 9: **end if**
 - 10: Update the sensors using $\bar{a}$ by U times and obtain $\hat{r}$
 - 11: $\tau(\bar{a}) \leftarrow \tau(\bar{a}) + 1$
 - 12: $\eta(\bar{a}) \leftarrow \eta(\bar{a})(1 - \frac{1}{\tau(\bar{a})}) + \frac{\hat{r}}{\tau(\bar{a})}$
 - 13: **end while**
 - 14: $a^* \leftarrow \arg\min_a \eta(a)$
 - 15: **return** a^*
-

is $a^* := \arg\min_a \eta(a)$, is returned. The overall procedure of the MAX- k -UCB is shown in Algorithm 1.

The UCB index provides the exploration-exploitation trade-off to determine the best-performing arm, and is known to achieve an asymptotically optimal performance [31], [32] particularly when the performances between arms are not related to each other. However, we note that there is scope for improvement. For example, we empirically observe that for a fixed k , if $|\gamma^{(k)} - \gamma'| > |\gamma^{(k)} - \gamma''|$, then the average cost of (k, γ') is higher than the average cost of (k, γ'') . This suggests that the learning algorithm can be improved by exploiting the structure of the objective cost function. However, in practice, the structure of the cost function is commonly unknown a priori. Furthermore, the search space for determining $(k^*, \gamma^{(k^*)})$ is a mixed space because k is a discrete variable and γ is a continuous variable. This makes it difficult for us to adopt existing structured MAB approaches. In the following subsection, we develop a new optimization approach for determining $(k^*, \gamma^{(k^*)})$ in a continuous (or mixed) space and with stochastic outcomes.

B. Optimizing MAX- k policy with GPSO

To determine $(k^*, \gamma^{(k^*)})$, we exploit the PSO technique [33], [34], which is a population-based optimization method that assumes a smooth underlying function. PSO maintains a population of particles called a swarm, where each particle represents a potential solution. While moving around the search space, each particle evaluates its position and moves closer to the area in which the optimal solution might reside. Specifically, the movement of each particle is guided by the direction of the best position experienced by the particle itself, and by the direction of the best position experienced by all particles. The swarm of particles pursues an optimal solution as a group while repeatedly moving in the guided direction.

PSO has some fascinating aspects: (i) simple implementation, (ii) no use of the gradient information of the objective function, and (iii) capability to search in a mixed space. These

Algorithm 2 MAX- k -GPSO policy

Input: p_1, p_2, w, U , swarm size $|S|$, set of cells C

- 1: Create a swarm of size $|S|$
- 2: For all particle $s \in S$, initialize $\mathbf{x}_s$ randomly, $\mathbf{v}_s \leftarrow 0$, and $\mathbf{y}_s \leftarrow \mathbf{x}_s$
- 3: **while** stopping condition is not true **do**
- 4: **for** each $s \in S$ **do**
- 5: $\mathbf{x}_s \leftarrow \mathbf{x}_s + \mathbf{v}_s$
- 6: $x_{s1} \leftarrow \lfloor 0.5 + x_{s1} \rfloor$
- 7: Make U transmissions under MAX- k policy with $\mathbf{x}_s = (k, \gamma)$ for updating state values of sensors and obtain $\hat{r}$.
- 8: $V^h(\mathbf{x}_s) \leftarrow V^h(\mathbf{x}_s) + 1$
- 9: $\tilde{g}^h(\mathbf{x}_s) \leftarrow \tilde{g}^h(\mathbf{x}_s)(1 - \frac{1}{V^h(\mathbf{x}_s)}) + \frac{\hat{r}}{V^h(\mathbf{x}_s)}$
- 10: **if** $\tilde{g}^h(\mathbf{x}_s) < \tilde{g}^h(\mathbf{y}_s)$ **then**
- 11: $\mathbf{y}_s \leftarrow \mathbf{x}_s$
- 12: **end if**
- 13: **end for**
- 14: $s^* \leftarrow \operatorname{argmin}_s \{\tilde{g}^h(\mathbf{y}_s)\}$, and $\hat{\mathbf{y}} \leftarrow \mathbf{y}_{s^*}$
- 15: Calculate ρ using (13)
- 16: **for** $d \in \{1, 2\}$ **do**
- 17: Draw e_3 uniformly at random in range $(0, 1)$
- 18: $v_{s^*d} \leftarrow -x_{s^*d} + \hat{y}_d + wv_{s^*d} + \rho(1 - 2e_3)$
- 19: **end for**
- 20: **for** all $s \in S \setminus \{s^*\}$ and $d \in \{1, 2\}$ **do**
- 21: Draw e_1, e_2 uniformly at random in range $(0, 1)$
- 22: $v_{sd} \leftarrow wv_{sd} + p_1e_1[y_{sd} - x_{sd}] + p_2e_2[\hat{y}_d - x_{sd}]$
- 23: **end for**
- 24: **end while**
- 25: **return** $\hat{\mathbf{y}}$

advantages are particularly attractive for our problem of determining $(k^*, \gamma^{(k^*)})$ because PSO can exploit the structure of the cost function in a mixed space. However, the original PSO method cannot be directly applied to our problem because it is suitable for deterministic objective functions. Because the cost under the MAX- k policy with a fixed (k, γ) is randomly drawn following an unknown fixed distribution, we have stochastic outcomes and require multiple visits to a position to obtain an accurate estimation of its function value, which is difficult to achieve in a continuous search space. Therefore, we modify the original PSO algorithm to address these challenges.

We begin by re-designing the search space to collect statistical information in a continuous search space. For the bounded values of k and γ , we divide the search space into *cells* and create a grid. Thereafter, we collect the per-cell statistical information obtained from the particles' visits to the cell. The average value can be considered an approximation of the objective cost function. Except for the cell-level "value" integration, the movement of each particle is identical to that of the original PSO. We combine this modified PSO or GPSO with the MAX- k policy and denote it by MAX- k -GPSO. The overall procedure of the MAX- k -GPSO is shown in Algorithm 2.

For $k \in [N]$ and real-valued $\gamma \in [\gamma^{mul}, \gamma^{all}]$, we partition the range of γ into M intervals of $\mathcal{I}$, and divide the two-

dimensional search space (k, γ) into a grid with $N \times M$ cells. Each cell c includes an integer k_c (for k) and range $[\underline{\gamma}_c, \bar{\gamma}_c] \in \mathcal{I}$ (for γ). Let C be the set of all cells. We introduce the following two functions:

- $h : [N] \times \mathbb{R} \rightarrow C$ maps (k, γ) to cell c with $k = k_c$ and $\gamma \in [\underline{\gamma}_c, \bar{\gamma}_c]$.
- $g : C \rightarrow \mathbb{R}$ maps cell c to the expected average cost over cell $\mathbb{E}_c[R]$, where the expectation is over $\gamma \in [\underline{\gamma}_c, \bar{\gamma}_c]$.

We aim to determine the cell c^* that minimizes $g(c)$, under the assumption that $g(c^*)$ is close to $g(h(k^*, \gamma^{(k^*)}))$. Because function $g(c)$ is initially unknown, we utilize empirical observations experienced by particles' visits to cell c while determining c^* . At time t , let $\tilde{g}_t(c)$ denote the empirical mean of the average costs obtained by visits to cell c up to time t . We omit subscript t if there is no confusion.

Assume we have a set S of particles in the swarm. The position of particle $s \in S$ is denoted by $\mathbf{x}_s = (x_{s1}, x_{s2})$ with $x_{s1} \in [N]$ and $x_{s2} \in [\gamma^{mul}, \gamma^{all}]$. Initially, we randomly locate the particles in the search space. Subsequently, for each particle s , we *evaluate for position* $\mathbf{x}_s$ during a round of U times: (i) we fix $(k, \gamma) = (x_{s1}, x_{s2})$, (ii) make U transmissions to update state values under the MAX- k policy (except that all sensor values are updated at the end of the round), and (iii) record the average cost for cell $h(\mathbf{x}_s)$, where we compute the average cost as (10). We then compute the empirical mean of the average cost of each cell as follows: Consider the number of visits of any particle to cell c up to round j as

$$V_j(c) = \sum_{\tau=1}^j \mathbb{I}_{\{h(\mathbf{x}^\tau)=c\}},$$

where $\mathbb{I}_{\{A\}}$ is the indicator function of event A ; $\mathbb{I}_{\{A\}} = 1$ if A is true and 0 otherwise, and $\mathbf{x}^\tau$ is the position of the particle that is evaluated at round τ . Initially, we set $V_0(c) = 0$ for all c . The empirical mean $\tilde{g}_j(c)$ for cell c is given by

$$\tilde{g}_j(c) = \frac{1}{V_j(c)} \sum_{\tau=1}^j \mathbb{I}_{\{h(\mathbf{x}^\tau)=c\}} \cdot \hat{r}_\tau,$$

where $\hat{r}_\tau$ is the empirical average cost obtained at the end of round τ . For all particles, we evaluate their position and compute the average cost of the visited cells (lines 4–13 in Algorithm 2).

We denote a group of $|S|$ rounds as a phase in which the position of each particle $s \in S$ has been evaluated for exactly one round of updates. After phase l , we now make the particles move. Let $\mathbf{y}_s$ denote the *local best position* of particle s based on its visit history and $\hat{\mathbf{y}}$ denote the *global best position* of all particles based on all the visit histories. They can be formally defined as

$$\begin{aligned} \mathbf{y}_s(l) &= (y_{s1}, y_{s2}) = \operatorname{argmin}_{\mathbf{x} \in \{\mathbf{x}_s(l), \mathbf{y}_s(l-1)\}} \tilde{g}^h(\mathbf{x}), \\ \hat{\mathbf{y}}(l) &= (\hat{y}_1, \hat{y}_2) = \operatorname{argmin}_{\mathbf{x} \in \{\mathbf{y}_s(l)\}_{s \in S}} \tilde{g}^h(\mathbf{x}), \end{aligned}$$

where $\mathbf{x}_s(l)$ is the location of particle s in phase l , and $\tilde{g}^h(\mathbf{x})$ denotes the composite function $\tilde{g}(h(\mathbf{x}))$. These two variables are used to compute the *velocity* $\mathbf{v}_s(l+1) = (v_{s1}(l+1), v_{s2}(l+1))$ of particle s at phase $l+1$, which is used to calculate the new position of particle s as

$$\begin{aligned} x_{s1}(l+1) &= \text{the closest integer to } (x_{s1}(l) + v_{s1}(l+1)), \\ x_{s2}(l+1) &= x_{s2}(l) + v_{s2}(l+1). \end{aligned}$$

The velocity is computed separately for each dimension. In this case, for each $d \in \{1, 2\}$, we have

$$v_{sd}(l+1) = w \cdot v_{sd}(l) + p_1 \cdot e_1 \cdot [y_{sd}(l) - x_{sd}(l)] + p_2 \cdot e_2 \cdot [\hat{y}_d(l) - x_{sd}(l)], \quad (11)$$

where the inertia weight w and acceleration coefficients p_1, p_2 are constants, and e_1 and e_2 are independent random variables drawn uniformly in the range $(0, 1)$. When the new position $\mathbf{x}_s(l+1)$ is determined for all s particles, the new phase $l+1$ begins from the evaluation of the position of each particle (i.e., a phase consisting of $|S|$ rounds).

The above behaviors are based on the original PSO algorithm, which can be possibly stuck in a suboptimal solution. We remedy this problem by modifying the movement of the best-performing particles, as suggested in [35]. At each phase l , let s^* denote the best-performing particle satisfying $\mathbf{y}_{s^*}(l) = \hat{\mathbf{y}}(l)$, and we call s^* the *global best particle*. The velocity of particle s^* is computed as $v_{s^*d}(l+1) = -x_{s^*d}(l) + \hat{y}_d(l) + wv_{s^*d}(l) + \rho(l)(1 - 2e_3)$, which implies that

$$x_{s^*d}(l+1) = \hat{y}_d(l) + wv_{s^*d}(l) + \rho(l)(1 - 2e_3), \quad (12)$$

where e_3 is another independent random variable drawn uniformly in the range $(0, 1)$, and $\rho(l)$ is determined as follows:

$$\rho(l+1) = \begin{cases} 2\rho(l), & \text{if } \mathbf{N}_{succ} > \lambda_s, \\ 0.5\rho(l), & \text{if } \mathbf{N}_f > \lambda_f \text{ and } \rho(l) > \underline{\rho}, \\ \rho(l), & \text{otherwise,} \end{cases} \quad (13)$$

where $\underline{\rho}, \lambda_s$, and λ_f are pre-determined parameters. Letting a "success" denote the case when $\tilde{g}^h(\hat{\mathbf{y}}(l+1)) < \tilde{g}^h(\hat{\mathbf{y}}(l))$ and a "failure" for the other case, $\mathbf{N}_{succ}$ is the number of phase-consecutive successes and $\mathbf{N}_{fail}$ is the number of phase-consecutive failures. The new velocity causes particle s^* to move to a point that is uniformly sampled from a square region with side lengths $2\rho(l)$ centered around $\hat{\mathbf{y}}(l) + w\mathbf{v}_{s^*}(l)$, where the region size depends on the event history of the successes and failures. As we will observe later, this behavior contributes to the convergence of the algorithm.

C. Convergence of Grid-PSO

We analytically demonstrate the convergence of the proposed GPSO algorithm to an optimal solution under the assumption that $g(c)$ is convex. The difficulty of the analysis originates from a lack of knowledge about the underlying objective function and its stochastic nature.

It has been shown in [36]–[38] that all particles converge to a point $\hat{\mathbf{y}}$ in the search space with appropriate parameters of w, p_1 , and p_2 , i.e.,

$$\lim_{l \rightarrow \infty} \mathbf{x}_s(l) = \hat{\mathbf{y}} \text{ for all } s \in S. \quad (14)$$

In the following, we show that if the converging point $\hat{\mathbf{y}}$ is not the optimal solution, then the particles can move toward the optimal solution.

Next, we generalize our formulation to provide results in a more general space. We consider the D -dimensional search space, and let $d \in [D]$ denote the index of the dimensions. In

each dimension d , we divide the search space into M_d exclusive intervals of identical length $2\delta_d$. Let $\delta_{max} := \max_d \delta_d$. Subsequently, the cell is hyper-rectangular with a side length $2\delta_d$ for each dimension d . We assume that the average cost $\hat{r}$ for each round is normalized to the range $[0, 1]$. For given $\epsilon > 0$, let G_ϵ^* be the set of cells whose average cost is ϵ -close to the minimum g^* where $g^* := \min_c g(c)$, i.e.,

$$G_\epsilon^* = \{c \mid g(c) - g^* < \epsilon\}.$$

Theorem 1. If average cost function $g(c)$ is convex, then for any $\epsilon > 0$, the particles converge to an optimal region under the GPSO algorithm, i.e., with probability 1,

$$\lim_{l \rightarrow \infty} h(\mathbf{x}_s(l)) \in G_\epsilon^*, \text{ for all } s \in S.$$

Proof. Suppose that $h(\hat{\mathbf{y}}(l)) \notin G_\epsilon^*$ at phase l . From (14), it is sufficient to show that the probability that the global best particle s^* moves to a position with smaller average cost in the next phase is greater than 0.

Let the largest side length of G_ϵ^* in dimension d be $2\sigma_d$, and $\sigma_{max} := \max_d \sigma_d$. Clearly, $\delta_{max} \leq \sigma_{max}$. We set GPSO with $\underline{\rho} = \sigma_{max}$. Let $H(l)$ denote the hypercube that particle s^* can move around, i.e., $\mathbf{x}_{s^*}(l+1) \in H(l)$. Let $\bar{\mathbf{y}}(l)$ denote the center of $H(l)$, i.e., $\bar{\mathbf{y}}(l) = \hat{\mathbf{y}}(l) + w\mathbf{v}_{s^*}(l)$. Because $H(l)$ has side length of $2\rho(l)$, our setting $\underline{\rho} = \sigma_{max}$ causes any side length of $H(l)$ to be always greater than the largest side length of G_ϵ^* . Subsequently, from the convexity of $g(\cdot)$ and the definition of G_ϵ^* , there exists $\mathbf{x} \in H(l)$ such that $\max |g(\mathbf{x}) - g(\bar{\mathbf{y}}(l))| > \epsilon$. We consider two cases: when the center $\bar{\mathbf{y}}(l)$ belongs to a cell in G_ϵ^* , i.e., when $h(\bar{\mathbf{y}}(l)) \in G_\epsilon^*$, and when it does not, i.e., $h(\bar{\mathbf{y}}(l)) \notin G_\epsilon^*$. In the following, we use the notation of the composite function $g^h(\mathbf{x}) := g(h(\mathbf{x}))$, and omit the subscript s^* for brevity.

Case 1 (when $h(\bar{\mathbf{y}}(l)) \in G_\epsilon^*$): There is always a positive probability that the next position of particle s^* is in a cell in G_ϵ^* since $\mathbf{x}(l+1)$ is uniformly sampled in $H(l)$. Hence, for certain ε_1 , we have

$$\text{Prob}\{h(\mathbf{x}(l+1)) \in G_\epsilon^*\} = \varepsilon_1 > 0.$$

Assume that particle s^* moves into a cell in G_ϵ^* . The global best position $\hat{\mathbf{y}}$ is set to $\mathbf{x}(l+1)$ if the empirical average $\tilde{\mu}' := \tilde{g}^h(\mathbf{x}(l+1))$ is less than $\tilde{\mu}'' := \tilde{g}^h(\hat{\mathbf{y}}(l))$. Additionally, let the true average cost of the two cells be $\mu' := g^h(\mathbf{x}(l+1))$ and $\mu'' := g^h(\hat{\mathbf{y}}(l))$, which satisfies $\mu' < \mu''$ since $h(\mathbf{x}(l+1)) \in G_\epsilon^*$ and $h(\hat{\mathbf{y}}(l)) \notin G_\epsilon^*$. We consider the following two events E_1 and E_2 for a constant $\bar{\epsilon} > 0$:

$$E_1 := \{\tilde{\mu}' < \mu' + \bar{\epsilon}\}, \text{ and } E_2 := \{\tilde{\mu}'' > \mu'' - \bar{\epsilon}\}.$$

From the i.i.d observations of average cost and their bounded support, we can use Hoeffding's inequality [39], and obtain

$$\text{Prob}\{E_1\} \geq 1 - e^{-2\bar{\epsilon}^2 V^h(\mathbf{x}(l+1))},$$

$$\text{Prob}\{E_2\} \geq 1 - e^{-2\bar{\epsilon}^2 V^h(\hat{\mathbf{y}}(l))},$$

where $V^h(\mathbf{x}(l+1))$ and $V^h(\hat{\mathbf{y}}(l))$ are the number of visits of the particles to the cell that $\mathbf{x}(l+1)$ and $\hat{\mathbf{y}}(l)$ belong, respectively. Letting ε_2 denote the probability that both events E_1 and E_2 occur, we have $\varepsilon_2 > 0$. This implies $\tilde{\mu}' < \tilde{\mu}''$;

thus, we have $\hat{\mathbf{y}}(l+1) \leftarrow \mathbf{x}(l+1)$. Hence, with probability $\varepsilon_1 \varepsilon_2 > 0$, the global best position $\hat{\mathbf{y}}$ changes to $\mathbf{x}(l+1)$ and we obtain $h(\hat{\mathbf{y}}(l+1)) \in G_\epsilon^*$.

Case 2 (when $h(\bar{\mathbf{y}}(l)) \notin G_\epsilon^*$): We note that, if we are stuck at a fixed $\hat{\mathbf{y}}(l)$, position $\mathbf{x}$ of particle s^* stays around $\hat{\mathbf{y}}(l)$ from (12), which implies that $\mathbf{v}(l) \rightarrow 0$. Thus, $\bar{\mathbf{y}}(l) \rightarrow \hat{\mathbf{y}}(l)$.

We now compare the average cost of $h(\mathbf{x}(l+1))$ with that of $h(\bar{\mathbf{y}}(l))$. From the convexity of $g(\cdot)$ and $\underline{\rho} = \sigma_{max}$, we should have

$$\min_{\mathbf{x} \in H(l)} g^h(\mathbf{x}) + \epsilon < g^h(\bar{\mathbf{y}}(l)).$$

Considering that the new position of the particle is chosen at random from $H(l)$, the equation implies that there is a positive probability ε_3 such that the particle moves to a better position.

$$\text{Prob}\{g^h(\mathbf{x}(l+1)) < g^h(\bar{\mathbf{y}}(l))\} = \varepsilon_3 > 0.$$

Since the best position $\hat{\mathbf{y}}$ can be updated only when the empirical average cost improves, as in Case 1, we can use the Hoeffding's inequality to show that the conditional probability of $\hat{\mathbf{y}}(l+1) \leftarrow \hat{\mathbf{x}}(l+1)$ is no smaller than $\varepsilon_4 = (1 - e^{-2\varepsilon^2 V^h(\mathbf{x}(l+1))})(1 - e^{-2\varepsilon^2 V^h(\bar{\mathbf{y}}(l))})$.

Combining together, with the positive probability $\varepsilon_3 \varepsilon_4 > 0$, $\hat{\mathbf{y}}(l)$ is set to a better position $\mathbf{x}(l+1)$ that satisfies $g^h(\mathbf{x}(l+1)) < g^h(\hat{\mathbf{y}}(l))$, which completes the proof. ■

From Theorem 1, MAX- k -GPSO policy can successfully find the cell in G_ϵ^* for any $\epsilon > 0$ under a convex average cost function $g(c)$. By using sufficiently small $\epsilon > 0$, it can achieve near-optimal solution. In the next section, we evaluate MAX- k -GPSO policy and compare its performance with multi-threshold, all-at-once, MAX- k -UCB policies.

V. SIMULATIONS

For the simulations, we consider a scenario where a system aims to estimate locations of N moving objects, each of which randomly walks along one-dimensional path. It captures the essentials of the system operations and its extension to multi-dimensional space is straightforward. Accordingly, the state value of each sensor corresponds to the location of a moving object and the estimation error corresponds to the distance between the true location and the estimated location.

We conducted simulations with N sensors, each of which has a state value that follows a standard Wiener process independently. We discretized time to simulate Wiener process evolution in simulations, and state values had increments that follow a normal distribution with mean 0 and variance 1 at each time slot. At each time t , the transmitter made an update decision $\{u_t^i\}_{i=1}^N$ and transmitted the state value of sensor i if $u_t^i = 1$. We assumed that a communication cost is imposed according to the number of transmitted state values using (2). Accordingly, the estimation errors $\mathcal{E}_i(t)$'s were updated on the receiver side. The system cost at time t is $\sum_{i=1}^N \mathcal{E}_i^2(t) + f(n_t)$, and the average system cost during time period was calculated as in (9). We divided the time into rounds, where a round continued for U consecutive transmissions of the state values. Letting t' and t'' be the starting and ending times of a round, the average cost $\hat{r}$ during a round was computed as

$$\frac{1}{t''-t'+1} \sum_{t=t'}^{t''} \left(\sum_{i=1}^N \mathcal{E}_i^2(t) + f(n_t) \right). \quad (15)$$

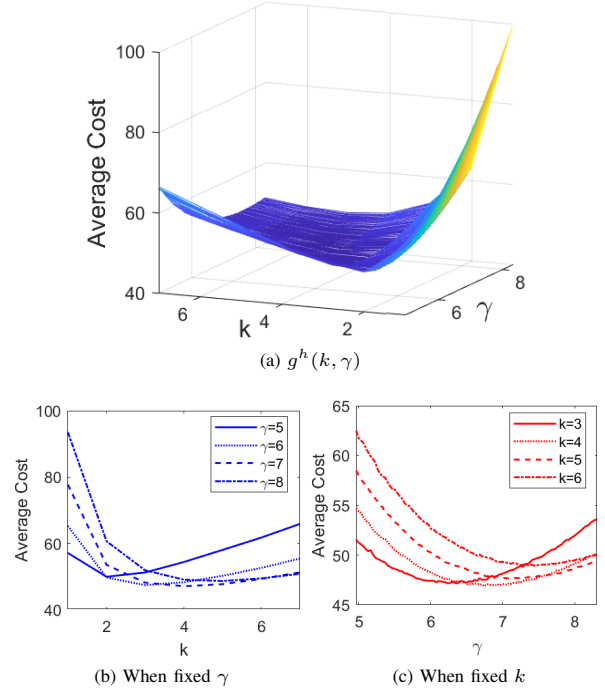


Fig. 5. Empirical observations on the objective function g .

For the online learning policies of MAX- k -GPSO and MAX- k -UCB, $n_{t''} = N$ since we reset the estimation errors to zero at the end of each round. Unless otherwise stated, we set parameters $U = N^2$ and $N = 7$, and used the communication cost function with $\alpha = 0.5$, and $\bar{c} = 100$. For the MAX- k -GPSO policy, we partitioned the search space for (k, γ) into the NM cells. We also considered NM arms for the MAX- k -UCB policy, each of which represents (k, γ) where $k \in [N]$ and $\gamma \in \{\gamma_1, \dots, \gamma_M\}$. When comparing the performance of the MAX- k -GPSO and MAX- k -UCB policies, we used the same constant value M .

First, we empirically observed the structure of the objective function g . We partitioned the range for γ into $M = 100$ intervals, each of which was centered at $\{\gamma_m\}_{m=1}^M$. We considered $NM = 700$ (k, γ) points, where $k \in [N]$ and $\gamma \in \{\gamma_1, \dots, \gamma_M\}$; for each (k, γ) , we calculated the average cost during 1000 rounds, i.e., $\frac{1}{1000} \sum_{j=1}^{1000} \hat{r}_j$. The average cost for the corresponding k and γ values is plotted in Fig. 5a. We can empirically observe the convex structure of the objective function g , and Figs. 5b and 5c show the functions for fixed γ and k , respectively. From these results, we expected that the MAX- k -GPSO policy successfully determined the optimal solution to our problem.

For the MAX- k -GPSO policy, we created a swarm of $|S|$ particles. According to [40], we set $|S| = 10 + \lfloor 2\sqrt{D} \rfloor$ where D is the number of dimensions in the search space. In our case of $D = 2$, $|S| = 12$. We also set the inertia weight and acceleration coefficients in (11) as $w = 0.7298$ and $p_1 = p_2 = 1.49618$ as in [41]. We partitioned the search space by dividing the range of γ into M exclusive intervals of length 2δ , resulting in a total NM cells. For the velocity equation of the global best particles s^* , we set $\underline{\rho} = 4\delta$, $\rho(0) = 1$, $\lambda_s = 5$, and $\lambda_f = 5$ in (13). For the MAX- k -UCB policy,

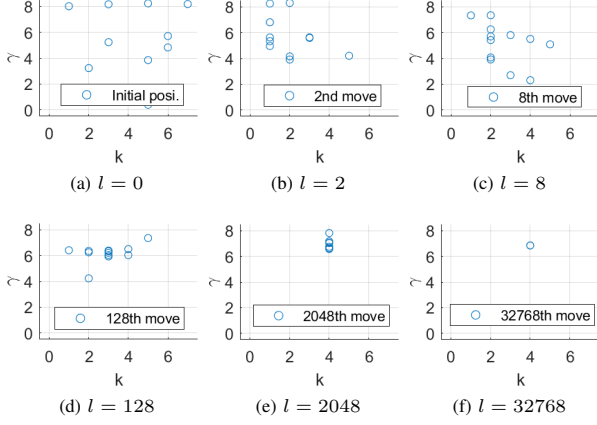


Fig. 6. Positions of particles ($|S| = 12$), after l -th phase.

we considered each (k, γ) as an arm with $k \in [N]$ and $\gamma \in \{\gamma_1, \dots, \gamma_M\}$, where γ_m is the center of the m -th interval. For both policies, we used the same stopping condition, running for J rounds. Fig. 6 shows the positions of all particles under the MAX- k -GPSO policy after l -th improvement phase when J was set to a million and $M = 1000$. Initially, the particles were randomly scattered over the search space. As the process continued, they moved toward the direction of the optimal solution and converged to the optimal solution of $k^* = 4$ and $\gamma^{(k^*)} = 6.8530$.

We now compare the learning performances of the MAX- k -GPSO and MAX- k -UCB policies. We measured their *regret* performance, a widely used performance metric in the learning area. This is defined as the accumulated cost sum with respect to the optimal average cost $\hat{r}^*$. Specifically, regret $reg_{j'}$ and average regret $\overline{reg}_{j'}$ in round j' can be formally expressed as

$$reg_{j'} = \sum_{j=1}^{j'} (\hat{r}_j - \hat{r}^*), \text{ and } \overline{reg}_{j'} = \frac{1}{j'} reg_{j'},$$

respectively. We empirically obtained $\hat{r}^*$ by running the MAX- k policy with parameters set to the convergence point of the MAX- k -GPSO policy. Fig. 7 shows the regret and average regret for $J = 10^6$ rounds, where the orange lines denote the regret performance of the MAX- k -UCB policy and the green lines denote that of the MAX- k -GPSO policy. We also simulated different cell sizes of $M = 100, 500, 1000$, and the results are shown by dotted, dashed, and solid lines, respectively. By comparing values of regrets in Fig. 7, we observe that the MAX- k -GPSO policy significantly outperforms MAX- k -UCB; even the MAX- k -GPSO policy with the finest grid converges faster than the MAX- k -UCB with the coarsest grid. Figs. 7c and 7d confirm that the MAX- k -GPSO with a coarser grid converged faster. In addition, we observed that, for MAX- k -GPSO with $M = 1000$, after the regret converged at approximately 25,000 rounds, it continued to increase gradually, which was due to the approximation error by discretizing the search space into cells.

In Fig. 8, we compare the average cost of the all-at-once and multi-threshold policies with that of the MAX- k -GPSO policy. The solid lines represent the ratio of the average cost of the all-at-once policy to that of the MAX- k -GPSO policy, and the

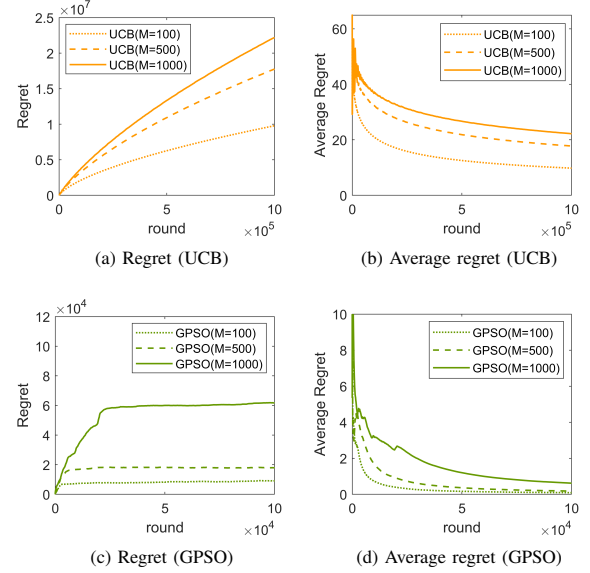


Fig. 7. Regret performance of MAX- k -GPSO & MAX- k -UCB.

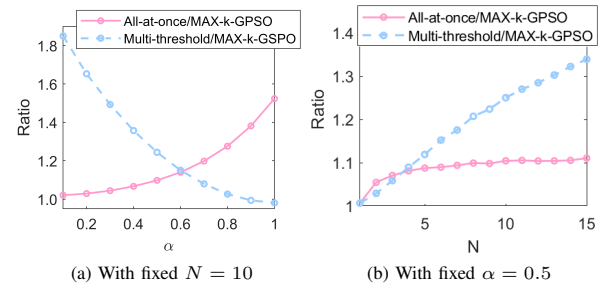


Fig. 8. Ratio of average cost.

dashed lines represent the ratio for the multi-threshold policy. The average cost was calculated for $J = 10^6$ rounds, and we set $M = 100$ for the MAX- k -GPSO policy. Fig. 8a shows the ratios with fixed $N = 10$ and different $\alpha \in [0.1, 1]$ of the cost function (2). The average cost under the all-at-once policy was greater than that under the MAX- k -GPSO policy for all α , and the ratio approached 1 when α approached 0. This was because the all-at-once policy was optimal when $\alpha = 0$. Similarly, we can expect that the ratio for the multi-threshold approaches 1 as α approaches 1. However, we observed that the ratio is less than 1 when $\alpha > 0.8$. This was due to the sub-optimality of MAX- k -GPSO caused by cell partitioning. The performance gap can be reduced by decreasing ϵ and increasing M . Fig. 8b shows the ratios with fixed $\alpha = 0.5$ and different $N \in [15]$. The average cost ratio of the multi-threshold policy increased almost linearly as N increased, whereas the ratio of the all-at-once policy increased sublinearly.

Through the simulations, we observed that the MAX- k -GPSO policy could find the optimal solution in our remote estimation problems, and it outperformed the all-at-once and multi-threshold policies for certain sublinear transmission cost functions.

VI. CONCLUSIONS

We investigated the remote tracking problem of monitoring multiple IoT sensors governed by Wiener processes under a sublinear communication cost. We first considered two competitive benchmark strategies, all-at-once and multi-threshold policies, and analyzed their performance. We then developed a novel strategy, the MAX- k policy, which could improve the average system cost as it better exploits the sublinear structure of the communication cost function. In its implementation, we blended a learning approach for the online optimization of its parameters. Specifically, we developed a GPSO that learns the parameters of MAX- k policy, considering the properties of continuous search space and stochastic feedback. Through simulations, we demonstrated that MAX- k -GPSO outperforms MAX- k -UCB, which combines a conventional UCB learning algorithm with the MAX- k policy. In addition, we demonstrated that the MAX- k -GPSO achieves close-to-optimal performance over different network settings, successfully generalizing the all-at-once and multi-threshold policies in extreme cases.

For the future works, we try to analyze the average cost under MAX- k policy and obtain optimal threshold values theoretically by understanding the behavior of k highest-valued Wiener processes. Further, we can investigate the optimal threshold under a hard constraint or dynamic network setting. Optimizing the threshold quickly to reduce the performance loss will be an interesting open problem.

REFERENCES

- [1] J. Yun, A. Eryilmaz, and C. Joo, "Remote tracking of dynamic sources under sublinear communication costs," in *IEEE INFOCOM Workshops*. IEEE, 2021, pp. 1–6.
- [2] C. Li, Y. Huang, S. Li, Y. Chen, B. A. Jalaian, Y. T. Hou, W. Lou, J. H. Reed, and S. Kompella, "Minimizing Aol in a 5G-Based IoT Network Under Varying Channel Conditions," *IEEE Internet of Things Journal*, vol. 8, no. 19, pp. 14543–14558, 2021.
- [3] O. S. Amosov and S. G. Amosova, "Peculiarities and Applications of Stochastic Processes with Fractal Properties," *Sensors*, vol. 21, no. 17, 2021.
- [4] M. Tang, S. Cai, and V. K. N. Lau, "Remote state estimation with asynchronous mission-critical iot sensors," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 3, pp. 835–850, 2021.
- [5] M. M. Rana, "Iot-based electric vehicle state estimation and control algorithms under cyber attacks," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 874–881, 2020.
- [6] G. M. Lipsa and N. C. Martins, "Remote state estimation with communication costs for first-order lti systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2013–2025, 2011.
- [7] A. Nayyar, T. Başar, D. Teneketzis, and V. V. Veeravalli, "Optimal strategies for communication and remote estimation with an energy harvesting sensor," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2246–2260, 2013.
- [8] O. C. Imer and T. Başar, "Optimal estimation with limited measurements," *International Journal of Systems, Control and Communications*, vol. 2, no. 1-3, pp. 5–29, 2010.
- [9] X. Gao, E. Akyol, and T. Başar, "Optimal communication scheduling and remote estimation over an additive noise channel," *Automatica*, vol. 88, pp. 57–69, 2018.
- [10] —, "On remote estimation with multiple communication channels," in *American Control Conference (ACC)*, 2016.
- [11] J. Chakravorty, J. Subramanian, and A. Mahajan, "Stochastic approximation based methods for computing the optimal thresholds in remote-state estimation with packet drops," in *American Control Conference (ACC)*, 2017.
- [12] J. Yun, C. Joo, and A. Eryilmaz, "Optimal real-time monitoring of an information source under communication costs," in *IEEE Conference on Decision and Control (CDC)*, 2018.
- [13] Y. Sun, Y. Polyanskiy, and E. Uysal-Biyikoglu, "Remote estimation of the wiener process over a channel with random delay," in *IEEE International Symposium on Information Theory (ISIT)*, 2017.
- [14] Y. Sun, Y. Polyanskiy, and E. Uysal, "Sampling of the wiener process for remote estimation over a channel with random delay," *IEEE Transactions on Information Theory*, vol. 66, no. 2, pp. 1118–1135, 2019.
- [15] K. Nar and T. Başar, "Sampling multidimensional wiener processes," in *IEEE Conference on Decision and Control*, 2014.
- [16] M. M. Vasconcelos and N. C. Martins, "Optimal estimation over the collision channel," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 321–336, 2016.
- [17] —, "Optimal remote estimation of discrete random variables over the collision channel," *IEEE Transactions on Automatic Control*, vol. 64, no. 4, pp. 1519–1534, 2018.
- [18] X. Zhang, M. M. Vasconcelos, W. Cui, and U. Mitra, "Distributed remote estimation over the collision channel with and without local communication," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 1, pp. 282–294, 2021.
- [19] V. Kostina and B. Hassibi, "Rate-cost tradeoffs in control," *IEEE Transactions on Automatic Control*, vol. 64, no. 11, pp. 4525–4540, 2019.
- [20] O. Sabag, P. Tian, V. Kostina, and B. Hassibi, "The minimal directed information needed to improve the lqg cost," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 1842–1847.
- [21] N. Guo and V. Kostina, "Optimal causal rate-constrained sampling for a class of continuous markov processes," *IEEE Transactions on Information Theory*, vol. 67, no. 12, pp. 7876–7890, 2021.
- [22] S. Kang, A. Eryilmaz, and C. Joo, "Comparison of Decentralized and Centralized Update Paradigms for Remote Tracking of Distributed Dynamic Sources," in *IEEE INFOCOM*, 2021.
- [23] S. Kang and C. Joo, "Index-based update policy for minimizing information mismatch with markovian sources," *Journal of Communications and Networks*, vol. 23, no. 6, pp. 488–498, 2021.
- [24] "802.11n: The next generation of wireless performance," White Paper, Cisco, Apr. 2009.
- [25] J. Hong, S. Park, H. Kim, S. Choi, and K. B. Lee, "Analysis of device-to-device discovery and link setup in lte networks," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2013, pp. 2856–2860.
- [26] J. Lorchat and T. Noel, "Energy saving in ieee 802.11 communications using frame aggregation," in *GLOBECOM'03. IEEE Global Telecommunications Conference (IEEE Cat. No. 03CH37489)*, vol. 3. IEEE, 2003, pp. 1296–1300.
- [27] D. Skordoulis, Q. Ni, H.-H. Chen, A. P. Stephens, C. Liu, and A. Jamalipour, "Ieee 802.11 n mac frame aggregation mechanisms for next-generation high-throughput wlangs," *IEEE Wireless Communications*, vol. 15, no. 1, pp. 40–47, 2008.
- [28] R. G. Gallager, *Stochastic processes: theory for applications*. Cambridge University Press, 2013.
- [29] J.-F. L. Gall, *Brownian Motion, Martingales, and Stochastic Calculus*. Springer, 2016.
- [30] O. C. Ibe, *Elements of Random Walk and Diffusion Processes*. John Wiley & Sons, 2013.
- [31] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2-3, May 2002.
- [32] S. Kang and C. Joo, "Low-complexity learning for dynamic spectrum access in multi-user multi-channel networks," in *IEEE INFOCOM*, April 2018.
- [33] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE ICNN*, vol. 4, 1995.
- [34] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *IEEE MHS*, 1995.
- [35] F. van den Bergh and A. P. Engelbrecht, "A new locally convergent particle swarm optimiser," in *IEEE International conference on systems, man and cybernetics*, vol. 3, 2002.
- [36] F. Van den Bergh and A. P. Engelbrecht, "A convergence proof for the particle swarm optimiser," *Fundamenta Informaticae*, vol. 105, no. 4, pp. 341–374, 2010.
- [37] R. Poli, "Mean and variance of the sampling distribution of particle swarm optimizers during stagnation," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 712–721, 2009.
- [38] C. W. Cleghorn and A. P. Engelbrecht, "Particle swarm convergence: Standardized analysis and topological influence," in *International Conference on Swarm Intelligence*. Springer, 2014, pp. 134–145.

- [39] W. Hoeffding, "Probability inequalities for sums of bounded random variables," in *The Collected Works of Wassily Hoeffding*. Springer, 1994, pp. 409–426.
- [40] M. Clerc, *Particle swarm optimization*. John Wiley & Sons, 2010, vol. 93.
- [41] F. Van Den Bergh and A. P. Engelbrecht, "An analysis of particle swarm optimizers," Ph.D. dissertation, 2002.

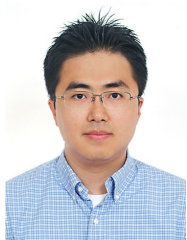


Jihyeon Yun received the MS degree from the School of ECE, Ulsan National Institute of Science and Technology (UNIST), in 2019. Currently, she studies in the Department of CSE, Korea University, toward the PhD degree. Her research interests include remote estimation, sensor networks, satellite networks, and reinforcement learning.



Atilla Eryilmaz (S'00 / M'06 / SM'17) received his M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign in 2001 and 2005, respectively. Between 2005 and 2007, he worked as a Postdoctoral Associate at the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology. Since 2007, he has been at The Ohio State University, where he is currently a Professor and the Graduate Studies Chair of the Electrical and Computer Engineering Department.

Dr. Eryilmaz's research interests span optimal control of stochastic networks, machine learning, optimization, and information theory. He received the NSF-CAREER Award in 2010 and two Lumley Research Awards for Research Excellence in 2010 and 2015. He is a co-author of the 2012 IEEE WiOpt Conference Best Student Paper, subsequently received the 2016 IEEE Infocom, 2017 IEEE WiOpt, 2018 IEEE WiOpt, and 2019 IEEE Infocom Best Paper Awards. He has served as: a TPC co-chair of IEEE WiOpt in 2014, ACM Mobihoc in 2017, and IEEE Infocom in 2022; an Associate Editor (AE) of IEEE/ACM Transactions on Networking between 2015 and 2019; an AE of IEEE Transactions on Network Science and Engineering between 2017 and 2022; and currently is an AE of the IEEE Transactions on Information Theory since 2022.



Jun Moon is currently an Associate Professor in the Department of Electrical Engineering at Hanyang University, Seoul, South Korea. He obtained his Ph.D. degree in electrical and computer engineering from the University of Illinois Urbana-Champaign, USA, in 2015. He received the B.S. degree in electrical and computer engineering and the M.S. degree in electrical engineering from Hanyang University, Seoul, South Korea, in 2006 and 2008, respectively. From Feb. 2008 to Jun. 2011, he was a Researcher at the Agency for Defense Development (ADD) in

South Korea. From Feb. 2016 to Feb. 2019, he was an Assistant Professor in the School of Electrical and Computer Engineering at Ulsan National Institute of Science and Technology (UNIST), South Korea. From Mar. 2019 to Aug. 2020, he was Assistant and Associate Professors in the School of Electrical and Computer Engineering at University of Seoul, South Korea. He was the recipient of the Fulbright Graduate Study Award 2011. His research interests include stochastic optimal control, nonlinear filtering, mean field games, distributed optimal control, networked control systems, and control of unmanned vehicles.



Changhee Joo received the Ph.D. degree from Seoul National University in 2005. He was with Purdue University and The Ohio State University, and then with Korea University of Technology and Education (KoreaTech) and Ulsan National Institute of Science and Technology (UNIST). Since 2019, he has been with Korea University. His research interests are in the broad areas of networking and machine learning. He was a recipient of the IEEE INFOCOM 2008 Best Paper Award, the KICS Haedong Young Scholar Award (2014), the ICTC 2015 Best Paper Award, and the GAMENETS 2018 Best Paper Award. He has served as a general chair of ACM MobiHoc 2022, an Associate Editor of the *IEEE/ACM Transactions on Networking* between 2016 and 2020, an Associate Editor of the *IEEE Transactions Vehicular Technology* since 2020, and a Division Editor of the *Journal of Communications and Networks* since 2020.