2018 IEEE Conference on Decision and Control (CDC)
Miami Beach, FL, USA, Dec. 17-19, 2018

Optimal sampling of multiple linear processes over a shared medium

Sebin Mathew, Karl H. Johansson, Aditya Mahajan

***Abstract*— In many emerging applications, multiple sensors transmit their measurements to a remote estimator over a shared medium. In such a system, the optimal sampling rates at each sensor depend on the nature of the stochastic process being observed as well as the available communication capacity. Our main contribution is to show that the problem of determining optimal sampling rates may be posed as a network utility maximization problem and solved using appropriate modifications of the standard dual decomposition algorithms for network utility maximization. We present two such algorithms, one synchronous and one asynchronous, and show that under mild technical conditions, both algorithms converge to the optimal rate allocation. We present a detailed simulation study to illustrate that the asynchronous algorithm is able to adapt the sampling rate to change in the number of sensors and the available channel capacity and is robust to packet drops.**

I. Introduction

Recent advances in wireless communication have enabled the use of wireless sensor networks in various applications such as environment monitoring, healthcare, home automation, and so on. Sensors in a wireless sensor network have limited power, computation, and communication capabilities and operate over a time-varying network. In applications where remote estimation is critical, each sensor periodically samples its observation and transmits the sampled measurements to a remote estimator over the network. When these sensors communicate over a shared medium, the available capacity becomes a limited resource that has to be allocated optimally to prevent congestion and minimize estimation error at the remote estimator.

Several variations of remote estimation and sensor scheduling under communication constraints have been investigated in the literature. A typical class of problems is to find optimal offline sensor schedule in terms of the estimation error covariance under various resource constraints [1]–[5]. Despite the advantage of low computation capacity requirement and simple implementation, offline methods work inefficiently. Event-based schedules, where a sensor communicates with the remote estimator only when a pre-defined event happens, often has superior performance compared to using time-based schedules [6], [7]. It has been shown that the optimal transmission policy for a scalar linear stochastic process either with limited transmission opportunities or with cost transmission is of the event-trigger form [8]–[10]. Sensor scheduling policies to achieve trade-off between sensor-to-estimator communication rate and the estimation quality for event-based sampling was investigated in [7], [11]–[13]

The traditional approach to monitor the system state is to sample and send the signals periodically. The problem of finding optimal time-periodic sensor schedules for estimating the state of discrete-time dynamical systems was discussed in [14]. An event-triggered control strategy was proposed in [15] by striking a balance with conventional periodic sampled-data control, leading to so-called periodic event-triggered control. Design trade-offs between estimation performance, processing delay, and communication cost for a sensor scheduling problem over a heterogeneous network was discussed in [16].

Resource allocation is a fundamental problem in shared communication networks. An efficient resource allocation strategy ensures successful sharing of the communication channel among sensors while maximizing system performance as a whole. On the other hand, stability, fairness and robustness of rate control algorithms is critical [17]. Resource allocation problems are typically framed as optimization problems where the objective is to maximize aggregate sensor utility over their transmission rates. The authors of [18] provide synchronous and asynchronous distributed algorithms for such a network utility maximization problem and prove their convergence in a static environment. Applying decomposition techniques allows us to identify critical information that needs to be communicated between nodes and across layers, and suggests how network elements should react in order to attain the global optimum. Decentralized techniques for utility-maximizing protocols – primal, dual, and cross decomposition was studied in [17]–[20].

The main contribution of this paper is the use of dual decomposition algorithms for optimal sampling in a remote estimation system. In Section II we discuss the system model and formulate the objective in terms of network utility of the remote estimation system. In Section III we describe the synchronous and asynchronous dual decomposition algorithms and prove their convergence. In Section IV we conduct a detailed simulation study and demonstrate the robustness of the asynchronous algorithm to changing network conditions and packet drops.

II. Model and problem formulation

A. Model

Consider a remote estimation system in which n sensors are connected to a remote estimator over a network. The sensors are indexed by the set $N := \{1, \dots, n\}$. Sensor i, $i \in N$,

Sebin Mathew and Aditya Mahajan are with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada (sebin.mathew@mail.mcgill.ca, aditya.mahajan@mcgill.ca). Karl H. Johansson is with ACCESS Linnaeus Centre, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden (kallej@kth.se).

observes a continuous-time stochastic process $\{X_i(t)\}_{t\geq 0}$, $X_i(t) \in \mathbb{R}$, where $X_i(0) \sim \mathcal{N}(0, \theta)$ and for $t \geq 0$

$$dX_i(t) = a_i X_i(t)dt + dW_i(t)$$

where $a_i \in \mathbb{R}$ is a known constant and $\{W_i(t)\}_{t\geq 0}$, $W_i(t) \in \mathbb{R}$, is a stationary stochastic process with finite variance and satisfies a technical assumption (A1) that we state later. We assume that the initial states $\{X_i(0)\}$, $i \in N$, and the processes $\{W_i(t)\}_{t\geq 0}$, $i \in N$, are independent. This implies that the stochastic processes $\{X_i(t)\}_{t\geq 0}$, $i \in N$, are independent across sensors.

Sensor i, $i \in N$, samples its observation at a rate of $R_i = 1/T_i$ and sends the sampled measurements to the remote estimator over the network. It is assumed that the network is ideal and does not cause any delays or packet drops.[1] However, the network has a finite capacity C and the rates $\mathbf{R} = (R_1, \dots, R_n)$ must lie in the rate region

$$\mathcal{R} = \left\{(R_1, \dots, R_n) \in \mathbb{R}^n_{\geq 0} : \sum_{i=1}^n R_i = C\right\}.$$

In between the sampling instances, the estimator generates estimates $\{\hat{X}_i(t)\}_{t\geq 0}$ to minimize the mean-squared error. In particular, it can be shown that the optimal estimation strategy is as follows: at the sampling time of sensor i, $\hat{X}_i(t) = X_i(t)$; at other times

$$d\hat{X}_i(t) = a_i \hat{X}_i(t)dt.$$

Define the error process as $E_i(t) = X_i(t) - \hat{X}_i(t)$. Then, for all sampling times $t = kT_i$, $k \in \mathbb{Z}_{\geq 0}$, $E_i(t) = 0$, and for $t \in (kT_i, (k+1)T_i)$, it follows the dynamics

$$dE_i(t) = a_i E_i(t)dt + dW_i(t).$$

Since $E_i(t)$ is a periodic process, the average mean-squared error (MSE) $M_i(T_i)$ for sensor i when the sampling period is T_i is given by

$$\begin{aligned} M_i(T_i) &= \frac{1}{T_i}\mathbb{E}\left[\int_0^{T_i} (X_i(t) - \hat{X}_i(t))^2 dt\right] \\ &= \frac{1}{T_i}\mathbb{E}\left[\int_0^{T_i} (E_i(t))^2 dt\right] \\ &= \frac{1}{T_i}\int_0^{T_i} \mathrm{Var}(E_i(t))dt. \end{aligned} \tag{1}$$

where $\mathbb{E}$ and Var denote mean and variance. The total mean-square error across all sensors is given by

$$\sum_{i=1}^n M_i(T_i). \tag{2}$$

Example 1 Suppose the noise process at sensor i, $i \in N$, is a Wiener process with variance σ_i^2. Then, the state process is a stationary Gauss-Markov (or a Ornstein-Uhlenbeck) process, which we denote by $\mathrm{GaussMarkov}(a_i, \sigma_i)$. For any $t \in (0, T_i)$, we have that

$$\mathbb{E}[E_i(t)] = 0, \quad \text{and} \quad \mathrm{Var}(E_i(t)) = \frac{\sigma_i^2}{2a_i}(e^{2a_i t} - 1).$$

Substituting this in (1), we get that

$$M_i(T_i) = \frac{\sigma_i^2}{2a_i}\left[\frac{e^{2a_i T_i} - 1}{2a_i T_i} - 1\right].$$

B. Problem formulation

We are interested in the following optimization problem.

Problem 1 *Find a rate vector* $(R_1, \dots, R_n)$ *in the rate region* $\mathcal{R}$ *that minimizes the mean-squared error* (2)*. Or formally,*

$$\min_{\mathbf{R}\in\mathbb{R}^n_{\geq 0}} \sum_{i=1}^n M_i\left(\frac{1}{R_i}\right) \text{ such that } \sum_{i=1}^n R_i \leq C. \tag{3}$$

The above problem is similar in spirit to the class of resource allocation problems known as network utility maximization [17]–[20]. We can think of the channel capacity as the resource and the total mean-squared error as the negative of the network utility. In network utility maximization, the constraint optimization problem (3) is viewed as a primal problem. Instead of solving it directly, one considers its Lagrangian dual:

$$D(\lambda) = \min_{\mathbf{R}\in\mathbb{R}^n_{\geq 0}} L(\mathbf{R}, \lambda)$$

where

$$\begin{aligned} L(\mathbf{R}, \lambda) &= \sum_{i=1}^n M_i\left(\frac{1}{R_i}\right) - \lambda\left(C - \sum_{i=1}^n R_i\right) \\ &= \sum_{i=1}^n \left(M_i\left(\frac{1}{R_i}\right) + \lambda R_i\right) - \lambda C. \end{aligned} \tag{4}$$

The key to a distributed solution lies in decomposing the dual objective (4) into two levels of optimization problems. In the context of the above model, at the lower level, it is assumed that the Lagrange multiplier λ is known and each sensor i, $i \in N$, chooses a sampling rate as a solution to the following optimization problem:

$$R_i^*(\lambda) = \arg\min_{R_i\in\mathbb{R}_{\geq 0}} M_i\left(\frac{1}{R_i}\right) + \lambda R_i. \tag{5}$$

At the higher level, the network assumes that the sensors use rates according to the solution of (5) and chooses the Lagrange multiplier by solving the dual problem

$$\min_{\lambda\in\mathbb{R}_{\geq 0}} L(\mathbf{R}^*(\lambda), \lambda), \tag{6}$$

where $\mathbf{R}^*(\lambda) = (R_1^*(\lambda), \dots, R_n^*(\lambda))$.

In the sequel, we consider two different algorithms for simultaneously solving (5) and (6); we call these *synchronous* and *asynchronous* algorithms. Both algorithms are iterative algorithms where the remote estimator updates the value of the Lagrange multiplier (or the shadow price) λ and sensor i updates the value of the rate R_i. In both algorithms it is assumed that the remote estimator can broadcast the Lagrange multiplier λ to all sensors. The synchronous and asynchronous algorithms differ in how the update of the rates R_i is communicated back to the remote estimator.

[1]Delays and packet drops will increase the mean-squared error but not change the nature of the problem in any fundamental way.

The synchronous algorithm is performed as part of the initial handshaking protocol during which the sensors can directly communicate their updated rates to the remote estimator. Thus, at each iteration, the remote estimator synchronously updates the rates of all sensors. The algorithm is described in detail in Section III-A.

The synchronous algorithm requires a control channel and additional bandwidth for signalling overhead. On the other hand, the asynchronous algorithm is an on line algorithm where no additional bandwidth is required for signalling. The sensors do not explicitly communicate the updated rates to the remote estimator. Rather they simply transmit data at the updated rates and the remote estimator infers the rate through the inter-arrival time between successive transmissions. The algorithm is described in detail in Section III-B.

For both algorithms, we assume that the model satisfies the following assumptions:

- **(A1)** For all sensors i, $i \in N$, and any sampling period $T_i \in \mathbb{R}_{\geq 0}$, the mean-squared error in (1) is strictly increasing and convex in T_i.
- **(A2)** $M_i(T_i)$ is twice differentiable and the curvature $M_i''(T_i)$ on $\mathcal{R}$ is uniformly bounded away from zero, i.e., there exists a positive constant $\bar{\gamma}$ such that $M_i''(T_i) \geq \bar{\gamma}$ for all $T_i \in \mathbb{R}_{\geq 0}$ and $i \in N$.

Since $R_i = 1/T_i$, we can equivalently write (A1) and (A2) as follows:

- **(A1')** For all sensors i, $i \in N$, and any sampling rate $R_i \in \mathbb{R}_{\geq 0}$, the mean-squared error in (1) is strictly decreasing and convex in R_i.
- **(A2')** $M_i(1/R_i)$ is twice differentiable and the curvature $M_i''(1/R_i)$ on $\mathcal{R}$ is uniformly bounded away from zero, i.e., there exists a positive constant $\bar{\gamma}$ such that $M_i''(1/R_i) \geq \bar{\gamma}$ for all $R_i \in \mathbb{R}_{\geq 0}$ and $i \in N$.

These assumptions are mild and will be satisfied in most sensor networks. In particular, if $\mathrm{Var}(E_i(t)) > ct^{\alpha}$ where c and α are positive constants, then $M_i(T_i)$ is strictly increasing in T_i; if $\mathrm{Var}(E_i(t))$ is differentiable, then, $M_i(T_i)$ is twice differentiable.

The above assumptions are satisfied in Example 1 when $a_i > 0$. To see that note that,

$$M_i'(T_i) = \frac{\sigma_i^2}{4a_i^2 T_i^2}\left[e^{2a_i T_i}(2a_i T_i - 1) + 1\right]$$

and

$$M_i''(T_i) = \frac{\sigma_i^2 e^{2a_i T_i} - 2M_i'(T_i)}{T_i}.$$

It can be shown that for all $x \in \mathbb{R}_{>0}$, $e^x(x-1)+1 > 0$ and $\frac{e^x}{2x} - \frac{e^x(x-1)+1}{x^3} > 1/6$. Substituting $x = 2a_i T_i$ with $a_i > 0$, we get that $M_i'(T_i) > 0$ and $M_i''(T_i) > 2a_i\sigma_i^2/3$, thus, (A1) and (A2) are satisfied.

III. The two dual-decomposition algorithms

A. The synchronous dual decomposition algorithm

The synchronous dual decomposition algorithm may be viewed as an iterative gradient descent algorithm used by the remote estimator to solve (6). The remote estimator starts

Algorithm 1 Synchronous allocation of sampling rates

Input Set of sensors $\mathcal{S}$, network capacity C, number of iterations K, gradient descent step size α, initial Lagrange multiplier λ_0.

```
procedure SYNC(S, C, K, α, λ_0)
    for k = 0 upto K do
        for each i in S do
            solve R_{i,k} : M_i'(1/R_{i,k}) − R_{i,k}^2 λ_k = 0
        end for
        λ_{k+1} = [λ_k − α(C − Σ_{i=1}^N R_{i,k})]^+
    end for
    return (R_{1,K}, ..., R_{n,K})
end procedure
```

with an initial guess λ_0 of the optimal Lagrange multiplier. At each iteration k, the following steps are performed (See Algorithm 1 for a formal description):

- Sensor i, $i \in N$, chooses a rate $R_{i,k} = R_{i,k}^*(\lambda_k)$ by solving the optimization problem (5), which is a convex optimization problem. One possible solution is to identify a rate $R_{i,k}$ that satisfies
$$M_i'(1/R_{i,k}) - \lambda_k R_{i,k}^2 = 0 \tag{7}$$
where M_i' denotes the derivative of M_i. Sensor i then communicates $R_{i,k}$ to the remote estimator.
- Upon receiving the updated rate vectors $(R_{1,k}, \dots, R_{n,k})$, the remote estimator updates the Lagrange multiplier in the direction of the gradient[2] as follows:
$$\lambda_{k+1} = \left[\lambda_k - \alpha_k\left(C - \sum_{i \in N} R_i^*(\lambda_k)\right)\right]^+, \tag{8}$$
where $\alpha_k > 0$ is an appropriately chosen learning rate and $[x]^+ = \max\{x, 0\}$.

Theorem 1 *Under assumptions* (A1') *and* (A2') *there exists a unique solution* $\mathbf{R}^* = (R_1^*, \dots, R_n^*)$ *of Problem 1. Moreover, for any initial guess* $\lambda_0 > 0$ *and sufficiently small step size* α*, the rates* $\mathbf{R}_k = (R_{1,k}, \dots, R_{n,k})$ *chosen in the synchronous dual decomposition algorithm converge to the optimal rates* $\mathbf{R}^*$ *as* $k \to \infty$.

Proof: Conditions (A1') and (A2') are equivalent to the conditions (C1) and (C2) of Theorem 1 in [18]. The proof follows from [18, Appendix I]. ■

B. The asynchronous dual decomposition algorithm

In the asynchronous dual decomposition algorithm, each sensor $i, i \in N$, keeps track of the time T_i^S at which it will take the next sample. The remote estimator keeps track of the times $T_{i,k}^R$ when each sensor last transmitted and its estimate of their transmission rates $\hat{R}_i$. The remote estimator initializes $T_{i,-1}^R = 0$ and chooses an initial guess λ_0 of

[2]Note that for a given choice of rates $\mathbf{R}$, the derivative of $L(\mathbf{R}, \lambda)$ with respect to λ is given by $C - \sum_{i=1}^n R_i$.

Algorithm 2 Asynchronous allocation of sampling rates

Input Network capacity C, Gradient descent step size α, initial Lagrange multiplier λ_0.

do initialization
  for each i in $\mathcal{S}$ **do**
    solve $R_i : M_i^{'}(1/R_i) - R_i^2\lambda_0 = 0$
    set $T_i^S = t + \frac{1}{R_i}$; $T_{i,-1}^R = t$, where $t =$ current time.
  end for
end

procedure ASYNC-SENSOR-i
  upon event $\langle$Current time $t = T_i^S\rangle$ **do**
    sample the state of the process and transmit
    observe updated λ
    solve $R_i : M_i^{'}(1/R_i) - R_i^2\lambda_0 = 0$
    set $T_i^S = T_i^S + (1/R_i)$
  end initialization
end procedure

procedure ASYNC-ESTIMATOR(C)
  upon event $\langle$Packet received for sensor $j_k\rangle$ **do**
    update $T_{i,k}^R$ as given in (9)
    update $\hat{R}_{i,k}$ as given in (10)
    $\lambda_{k+1} = \left[\lambda_k - \alpha(C - \sum_{i=1}^N \hat{R}_{i,k})\right]^+$
    $k = k + 1$
  end initialization
end procedure

the Lagrange multiplier and broadcasts it to all sensors. Each sensor $i, i \in N$, then computes R_i by solving (7) and initializes $T_i^S = 1/R_i$. Then the following steps are performed at every iteration k (See Algorithm 2 for formal description):

- Let j_k denote the sensor with the lowest sampling time T_i^S. At time $T_{j_k}^S$, sensor j_k takes a sample and sends its measurements to the remote estimator over the network.
- Upon receiving the message from sensor j_k, the remote estimator sets

$$T_{i,k}^R = \begin{cases} T_{j_k}^S & \text{if } i = j_k \\ T_{i,k-1}^R & \text{otherwise} \end{cases} \tag{9}$$

  and

$$\hat{R}_{i,k} = \begin{cases} \dfrac{1}{T_{i,k}^R - T_{i,k-1}^R} & \text{if } i = j_k \\ \hat{R}_{i,k-1} & \text{otherwise} \end{cases} \tag{10}$$

- The remote estimator then chooses λ_{k+1} by updating the Lagrange multiplier by taking a step in the direction of the gradient as follows:

$$\lambda_{k+1} = \left[\lambda_k - \alpha_k\left(C - \sum_{i \in N} \hat{R}_{i,k}(\lambda_k)\right)\right]^+, \tag{11}$$

  where $\alpha_k > 0$ is an appropriately chosen learning rate and $[x]^+ = \max\{x, 0\}$.

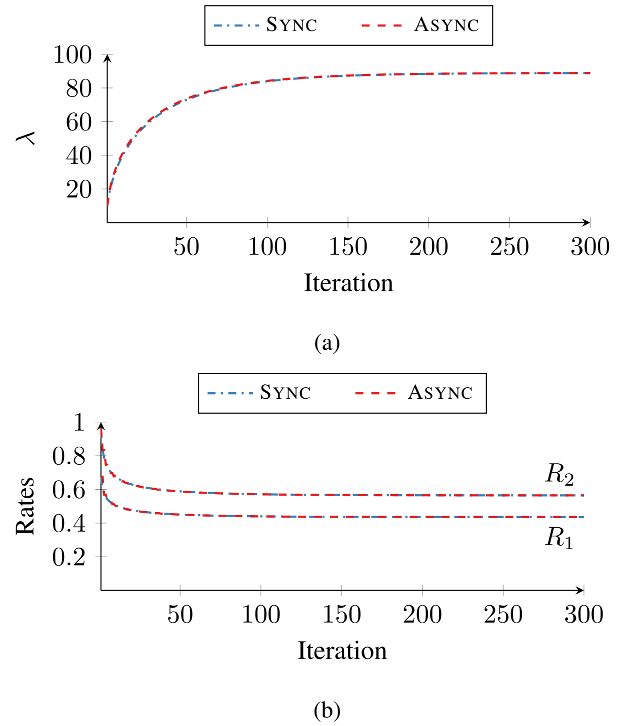


Fig. 1: Plot of (a) Lagrange multiplier λ and (b) rates R_1 and R_2 versus iteration for the illustrative example of Sec. III-C.

- The Lagrange multiplier λ_{k+1} is broadcast and sensor j_k updates the sampling rate R_{j_k} according to (5) or (7) and sets $T_{j_k}^S = T_{j_k}^S + \frac{1}{R_{j_k}}$.

Let $\mathcal{T}^R = \{T_{j_k,k}^R\}_{k \geq 0}$ denote the set of time instances at which the remote estimator updates the Lagrange multiplier based on the current estimate of the sensor rates. Also, let $\mathcal{T}_i^S$, $i \in N$, denote the set of time instances at which the sensor i updates its rate based on the Lagrange multiplier broadcast to it. It is assumed that the following is satisfied:

(A3) The time between consecutive updates in $\mathcal{T}^R$ (i.e., at the remote estimator) and $\mathcal{T}_i^S$, $i \in N$, (i.e., at every sensor) are bounded.

Note that (A3) is satisfied if for all $\lambda \in \mathbb{R}_{>0}$, the optimal rate $R_i^*(\lambda)$ obtained in (5) is bounded. This is the case in Example 1 if $a_i > 0$.

Theorem 2 *Under assumptions* (A1'), (A2') *and* (A3), *for any initial guess* $\lambda_0 > 0$ *and sufficiently small step size* α_k, *the rates* $\mathbf{R}_k = (R_{1,k}, \dots, R_{n,k})$ *chosen in the asynchronous dual decomposition algorithm converges to the unique solution* $\mathbf{R}^*$ *of Problem 1. Moreover, if the synchronous and asynchronous algorithms use the same learning rates* $\{\alpha_k\}_{k \geq 0}$, *then the corresponding Lagrange multipliers converge to the same value.*

Proof: Assumption (A3) being equivalent to (C3) in [18], the proof follows from [18, Theorem 2]. ■

Remark 1 The dual decomposition algorithm does not ensure that the dual iterates are feasible (i.e., at the intermediate

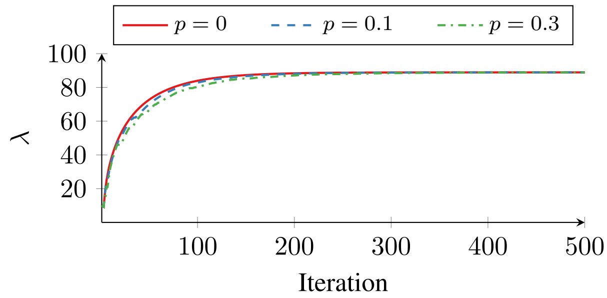


Fig. 2: Plot of Lagrange multiplier λ versus iteration for the asynchronous algorithm under packet drop for the illustrative example of Sec. III-C.

steps of the algorithm, it is not guaranteed that $\sum_{i\in N} R_{i,k} \leq C$). To keep the iterates feasible, one could start with a large value of λ_0, which will ensure that the sensors pick small values of the initial rates $\{R_i\}_{i\in N}$.

Remark 2 A feature of Algorithm 2 is that the clocks at the sensors and the remote estimator do not need to be synchronized.

C. An illustrative example

To illustrate how the algorithm works, consider a system with two sensors, $\mathrm{GaussMarkov}(1,1)$ and $\mathrm{GaussMarkov}(1,2)$, and a total capacity of $C = 1$. For the synchronous algorithm, suppose the remote estimator starts with an initial guess $\lambda_0 = 10$ and both sensors use a constant learning rate[3] of $\alpha_k = 10$. Then, the rates converge to $(R_1, R_2) = (0.4355, 0.5645)$ and $\lambda = 88.9136$. After 200 iterations, the value of λ is 88.357, which is within 0.625% of the optimal value.

For the asynchronous algorithm, we again assume that the remote estimator starts with an initial guess $\lambda_0 = 10$ and both sensors use a constant learning rate of $\alpha_k = 10$. After 200 iterations, the value of λ is 88.399, which is within 0.579% of the optimal value.

For comparison, we plot the value of the Lagrange multiplier λ and rates R_1 and R_2 vs iteration for both the synchronous and the asynchronous algorithms in Fig. 1. As can been seen from the figure, at each iteration, the Lagrange multiplier and the rates for both the synchronous and the asynchronous algorithms are fairly close. The key difference is that the synchronous algorithm is implemented as part of the initial handshaking protocol (which requires an additional signaling overhead) while the asynchronous algorithm is on line where the sensors adapt their transmission rates while transmitting data (so there is no signaling overhead). The 200 iterations of asynchronous algorithm takes about 194 sec.

D. Robustness of asynchronous algorithm to packet drops

For the asynchronous algorithm we assumed an ideal communication channel. The algorithm is robust to packet drops introduced by the channel as long as assumption (A3) continues to hold. To illustrate this point, we reconsider the example of Sec. III-C but assume that packets are dropped with probability p. The plot of Lagrange multiplier versus number of iterations for different values of packet drop probability p is shown in Fig. 2. As can be seen from the figure, there is very little impact of packet drops on the convergence of the algorithm.

IV. Numerical example

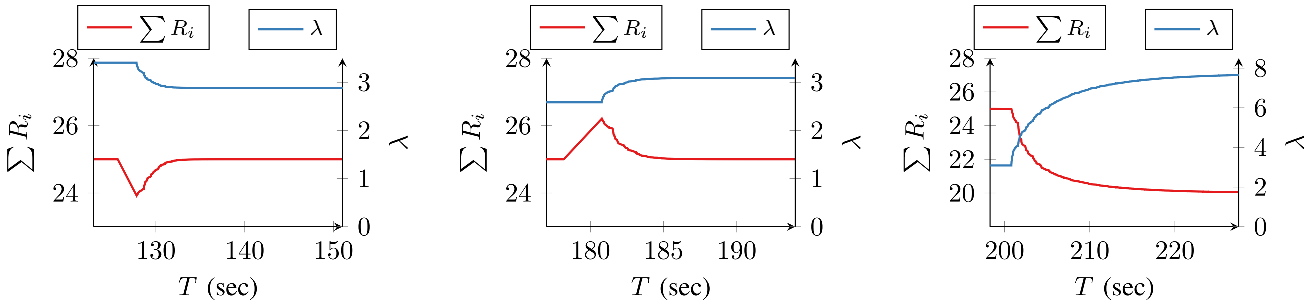
In the model described in Sec. II, the system is assumed to be static. However, in many applications, the network conditions change with time: new sensors may come onboard, existing sensors may leave, or the channel capacity might change. In general, the dual decomposition algorithm is robust to slow changes in the network conditions, so we expect the asynchronous algorithm to be able to adapt to changing network conditions. In this section, we present a detailed simulation study to illustrate the robustness of the asynchronous algorithm to network changes.

We consider an experimental setup where the number of sensors $N(t)$ changes according to a stochastic process. We assume that new sensors arive according to a Poisson process with rate ρ and stays in the system for an exponentially distributed amount of time with rate ρ, after which the sensor leaves the system. Each new sensor is $\mathrm{GaussMarkov}(a_i, \sigma_i)$ where a_i and σ_i are chosen randomly. We assume that the remote estimator broadcasts the value of the Lagrange multiplier λ at all times. When a new sensor arrives, its initial sampling rate is determined based on the current value of λ. The remote estimator continues to adapt λ according to Algorithm 2, without being explicitly aware that a new sensor has arrived. Similarly, the remote estimator is not explicitly aware when a sensor leaves the system.

We consider a scenario of 450 seconds where we start with $N(0) = 25$ sensors and sensors arrive and leave at a rate of $\rho = 2$ per minute. Each new sensor is $\mathrm{GaussMarkov}(a_i, \sigma_i)$, where $a_i \sim$ Unif$[0.1, 2]$ and $\sigma_i = 1$. At $T = 0$, the system capacity is 25; at $T = 200$, the capacity changes to $C = 20$; and at $T = 400$, it changes to $C = 30$. We run the asynchronous algorithm with a constant learning rate of $\alpha_k = 0.01$ throughout. The plot of $N(t)$ and λ versus time as well as $\sum_{i\in N} R_i$ versus time are shown in Fig. 4. These plots illustrate the robustness of the asynchronous algorithm to changing network conditions.

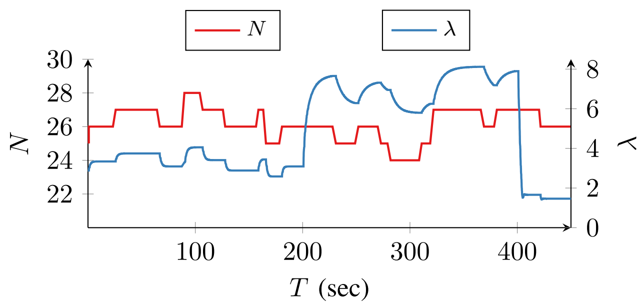
In Fig. 3(a), we zoom into Fig. 4(b) at $T = 127$ when the system has 27 sensors, $C = 25$, and $\lambda = 3.41$. At this time, one of the sensors leaves and the sum rate falls below the network capacity. The remote estimator adjusts the Lagrange multiplier λ according to (6). Since there is one less sensor

[3]In practice, convergence speeds up considerably if the learning rate is adapted according the gradient (e.g., using ADAM or ADAGrad [21]). However, in this example, we choose a constant learning rate to simplify exposition.

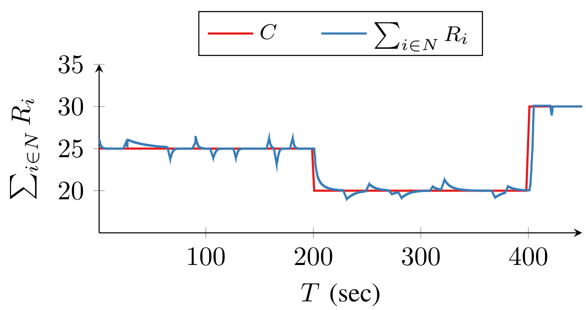


(a) (b) (c)

Fig. 3: Plot of sum rate $\sum_{i \in N} R_i$ and λ versus time for the asynchronous algorithm, illustrating (a) sensor leaving, (b) sensor coming aboard and (c) capacity change for the system described in Sec. IV.



(a)



(b)

Fig. 4: Plot of (a) number of sensors N and λ, and (b) sum rate $\sum_{i \in N} R_i$ versus time for the asynchronous algorithm for the system described in Sec. IV.

competing for the same resource, the Lagrange multiplier decreases and converges to 2.88 at $T = 135$.

In Fig. 3(b), we zoom into Fig. 4(b) at $T = 180$ when the system has 25 sensors, $C = 25$, and $\lambda = 2.58$. At this time, a new sensor comes aboard, sees the current value of λ and chooses a transmission rate using (5). When the new sensor transmits, the sum rate exceeds the channel capacity[4] The remote estimator adjusts Lagrange multiplier λ according to (6). Since there is one more sensor competing for the same resource, the Lagrange multiplier increases and converges to 3.09 at $T = 189$.

[4]In practice, the sum rate exceeding the channel capacity will result in delay or packet drops but such effects are not taken into account in our model.

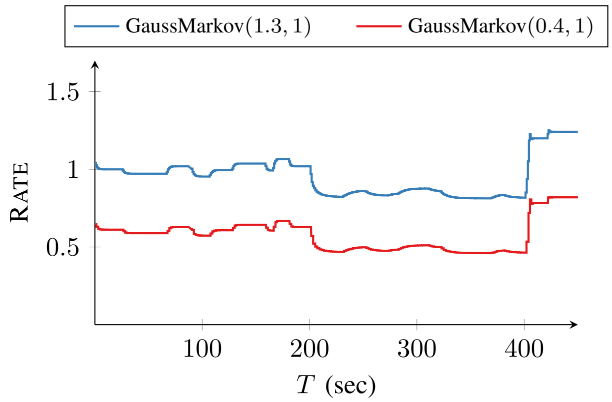
In Fig. 3(c), we zoom into Fig. 4(b) at $T = 200$ when the system has 26 sensors, $C = 25$, and $\lambda = 3.09$. At this time, the system capacity reduces to $C = 20$. The remote estimator adjusts the Lagrange multiplier using (6). Since there are the same number of sensors competing for less resources, the Lagrange multiplier increases and converges to 7.65 at $T = 227$.

To observe how individual sensor rates vary with changes in network conditions, we pick two sensors, $\text{GaussMarkov}(1.3, 1)$ and $\text{GaussMarkov}(0.4, 1)$ that stay active throughout the experiment. The rate allocation by the asynchronous algorithm for these sensors is shown in Fig. 5(a) and the empirical MSE is shown in Fig. 5(b).

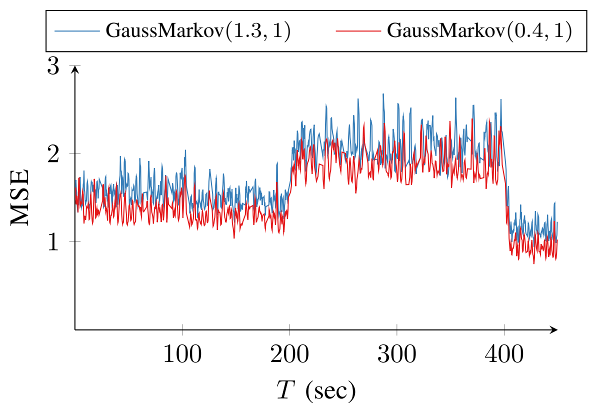
We compare the optimal rate allocation described in the asynchronous dual decomposition algorithm with two baseline equal rate allocation schemes. In Scheme 1, we assume that $N(t) \leq 30$ and allocate a constant sampling rate $R_i = C/30$ to all active sensors $i \in N$; in Scheme 2, we assume that the remote estimator keeps track of $N(t)$ and allocates a rate of $R_i = C/N(t)$ to all active sensors $i \in N$. We plot the aggregate empirical MSE for the system in Fig. 6, which shows that, as expected, the optimal scheme performance better than the two baselines and the difference in performance is significant when the channel capacity is low.

V. Conclusion

In this paper, we proposed a dual decomposition technique to minimize mean-squared error in a remote estimation system subject to capacity constraint, by posing the objective as a variant of the network utility maximization problem. We derived an asynchronous rate allocation algorithm where the sensors and the remote estimator communicate and update their controls asynchronously. The two dual decomposition algorithms described in the paper provide a decentralized approach to rate allocation for sensors communicating over a shared medium. The algorithms are provably convergent to the global optimum in a static network and robust to slowly changing network conditions and packet drops. Using an



(a)



(b)

Fig. 5: Plot of (a) sampling rate and (b) empirical MSE versus time for $\mathrm{GaussMarkov}(1.3, 1)$ and $\mathrm{GaussMarkov}(0.4, 1)$ for the system described in Sec. IV.

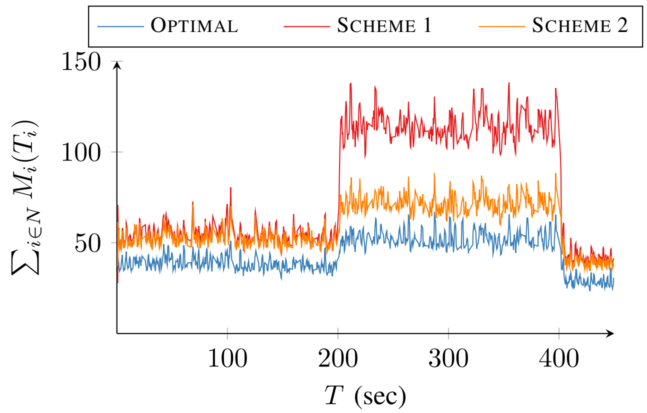


Fig. 6: Plot of aggregate empirical MSE illustrating optimality of the asynchronous algorithm against baseline rate allocation schemes 1 and 2, for the system described in Sec. IV.

experimental setup, the performance and optimality of the asynchronous algorithm is illustrated.

REFERENCES

[1] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray, "On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage," *Automatica*, vol. 42, no. 2, pp. 251–260, 2006.

[2] A. Tiwari, M. Jun, D. E. Jeffcoat, and R. M. Murray, "Analysis of dynamic sensor coverage problem using Kalman filters for estimation," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 53–58, 2005.

[3] C. Yang and L. Shi, "Deterministic sensor data scheduling under limited communication resource," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 5050–5056, 2011.

[4] L. Shi and H. Zhang, "Scheduling two Gauss-Markov systems: An optimal solution for remote state estimation under bandwidth constraint," *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 2038–2042, 2012.

[5] C. O. Savage and B. F. La Scala, "Optimal scheduling of scalar Gauss-Markov systems with a terminal cost function," *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 1100–1105, 2009.

[6] W. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," *Proceedings of the 51st IEEE Conference on Decision and Control (CDC)*, no. 3270–3285, 2012.

[7] J. Chakravorty and A. Mahajan, "Fundamental limits of remote estimation of autoregressive markov processes under communication constraints," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1109–1124, Mar. 2017.

[8] O. C. Imer and T. Basar, "Optimal estimation with limited measurements," *Proceedings of the 44th IEEE Conference on Decision and Control (CDC)*, pp. 1029–1034, 2005.

[9] 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.

[10] A. Molin and S. Hirche, "An iterative algorithm for optimal event-triggered estimation," in *4th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS'12)*, 2012.

[11] R. Cogill, S. Lall, and J. P. Hespanha, "A constant factor approximation algorithm for event-based sampling," *Proceedings of the American Control Conference (ACC)*, pp. 305–311, 2007.

[12] J. Wu, Q.-S. Jia, K. H. Johansson, and L. Shi, "Event-based sensor data scheduling: Trade-off between communication rate and estimation quality," *IEEE Transactions on automatic control*, vol. 58, no. 4, pp. 1041–1046, 2013.

[13] L. Li, M. Lemmon, and X. Wang, "Event-triggered state estimation in vector linear processes," *Proceedings of the American Control Conference (ACC), 2010*, pp. 2138–2143, 2010.

[14] S. Liu, M. Fardad, E. Masazade, and P. K. Varshney, "Optimal periodic sensor scheduling in networks of dynamical systems," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3055–3068, 2014.

[15] W. Heemels, M. Donkers, and A. R. Teel, "Periodic event-triggered control for linear systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 847–861, 2013.

[16] H. Sandberg, M. Rabi, M. Skoglund, and K. H. Johansson, "Estimation over heterogeneous sensor networks," in *Proceedings of 47th IEEE Conference on Decision and Control (CDC)*. IEEE, 2008, pp. 4898–4903.

[17] F. P. Kelly, A. K. Maulloo, and D. K. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research society*, vol. 49, no. 3, pp. 237–252, 1998.

[18] S. H. Low and D. E. Lapsley, "Optimization flow control. I. Basic algorithm and convergence," *IEEE/ACM Transactions on networking*, vol. 7, no. 6, pp. 861–874, 1999.

[19] B. Johansson, P. Soldati, and M. Johansson, "Mathematical decomposition techniques for distributed cross-layer optimization of data networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1535–1547, 2006.

[20] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 255–312, 2007.

[21] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.