Unifying AoI Minimization and Remote Estimation — Optimal Sensor/Controller Coordination with Random Two-way Delay

Cho-Hsin Tsai and Chih-Chun Wang, *Senior Member, IEEE*

***Abstract*—The ubiquitous usage of communication networks in modern sensing and control applications has kindled new interests on the timing coordination between sensors and controllers, i.e., how to use the "waiting time" to improve the system performance. Contrary to the common belief that a zero-wait policy is optimal, Sun *et al.* showed that a controller can strictly improve the data freshness, the so-called Age-of-Information (AoI), by postponing transmission in order to lengthen the duration of staying in a good state. The optimal waiting policy for the *sensor* side was later characterized in the context of remote estimation. Instead of focusing on the sensor and controller sides separately, this work develops the jointly optimal sensor/controller waiting policy in a Wiener-process system. This work generalizes the above two important results in the sense that not only do we consider joint sensor/controller designs (as opposed to sensor-only or controller-only schemes), but we also assume random delay in both the forward and feedback directions (as opposed to random delay in only one direction). In addition to provable optimality, extensive simulation is used to verify the performance of the proposed scheme.**

***Index Terms*—Age-of-information, remote estimation, optimal sampling, stochastic control, data freshness, information update system, infinite-horizon Markov decision process.**

I. Introduction

The omnipresence of portable devices has led to increasing focus on systems with multiple sensors and controllers interconnected by communication networks. Many new research directions have been initiated, including healthcare, energy management systems, cloud data infrastructure (see [1]–[3]). In this work, we study the question: How to optimally coordinate the sensor and the controller when there is random delay in both the forward and backward directions? We begin by observing there are two distinct ways of timing-based system optimization: *data-freshness control* and *state-based sampling*.

Data-freshness control: In this approach, the controller is the one that actively maintains the data-freshness of the system. For example, we consider the case where the goal is to lower the risk of heart attacks of the patients. One way is for the hospital (controller) to make sure that the blood pressure (BP) or the heart rate (HR) records of the patients are as fresh as possible. To this end, the hospital should intermittently request the patients (sensors) to measure their latest BP or HR and send in the reports. In practice, any sensor-to-controller measurement packet inevitably experiences some delay and is thus always "stale" to some degree. The controller (hospital) must decide how to optimize its request schedule in order to optimize the data-freshness of its records.

One breakthrough of the data-freshness control is the introduction of a new metric, Age-of-Information (AoI) [4], the corresponding minimization algorithms [5], and its numerous follow-up results [6]–[8]. For instance, a "generate-at-will" model was studied in [9], which has the potential of considerable energy savings.

In general, AoI minimization behaves differently from throughput maximization. For example, the zero-wait policy [10] was throughput optimal but can be strictly suboptimal in terms of the average AoI [11]. In [11] Sun *et al.* characterized the optimal "waiting time" policy at the controller side that can provably minimize the average AoI, i.e., the optimal policy when a hospital (controller) should request its patient (sensor) to submit his/her BP/HR report.

State-based sampling: Unlike the data-freshness control, in this line of research, it is the sensor that actively optimizes the overall system.[1] Continue from the aforementioned hospital-patient example. The state-based sampling approach is for the patient (sensor) to measure his/her own BP/HR continuously and report the value when and only when it shows elevated risk. Once the hospital (controller) receives the report, some treatment (action) is prescribed to bring the BP/HR back to normal. The patient will stay inactive afterwards and only send in new reports if his/her BP/HR starts to exhibit new concerns.

The focus of this direction is thus to design schemes that detect the changes in signal/measurement values and

This work was supported in parts by NSF under Grant CCF-1422997, Grant CCF-1618475, Grant CCF-1816013, and Grant CCF-2008527.

Part of this work was presented in 2020 IEEE International Conference on Computer Communications (INFOCOM).

The authors are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: chtsai@purdue.edu; chihw@purdue.edu).

[1] The best way to determine whether a scheme is *controller-based* or *sensor-based* is to examine in which physical location the decision is made, since their distinct locations naturally lead to *asymmetric access to the underlying random states and timing information*. Also see our discussion in Sec. II. However, such a definition does not apply to many existing results. The reason is that with the assumption of instantaneous ACK feedback, one node has complete and instantaneous access to the information available at the other node, which breaks the information asymmetry and thus blends the roles of sensors and controllers. The second way of classification is thus to see whether the algorithm has instantaneous access to the (random) value of the measurement and whether it explicitly uses the measurement to decide when to transmit. If so, it is a senor-side algorithm, e.g., the remote estimation scheme in [12]. Otherwise, it is a controller-side algorithm. Under this methodology, the AoI minimization scheme in [11] is classified as controller-based even though it is actually executed by the sensor. That is, one can envision "the controller" being a separate computer program within the physical sensor that tells the sensor when to transmit *without using the actual measurement data*.

opportunistically send the updates when the need arises. This direction is often termed the (state-based) sampling schemes for remote estimation. An early work [13] showed that a *threshold policy* can lower the estimation error. Later it was shown that the *threshold policy* is optimal for a variety of settings, including cellular networks [14], noisy channels [15] and multi-dimensional Wiener processes in [16]. In [12], Sun *et al.* generalized the setting of [16] by adding a queue with random service time between the sampler and estimator, and showed that the optimal waiting time at the sensor side again takes the form of a threshold policy. Further discussion of the threshold policy will be provided in Sec. IV.

The main motivation of this work is two-fold. Firstly, as shown in two important studies [11], [12], either controller (hospital) or sensor (patient) alone can significantly improve the system performance. One thus cannot help but wonder how much improvement one may experience with a jointly optimal sensor/controller policy. Secondly, since we are interested in remote systems with non-collocated sensors and controllers, there is likely to be random delay for both the sensor-to-controller and the controller-to-sensor directions [17]. Nonetheless, existing results [11] and [12] and all the aforementioned works assume random delay in one direction, plus idealized zero-delay acknowledgment (ACK) for the other direction. It is thus of paramount interest to study new optimal schemes under a more realistic 2-way delay model. Our key contributions are summarized as follows.

(i) We propose a new framework that unifies the controller-side AoI minimization problem [11] and the sensor-side remote estimation problem [12].

(ii) Our framework allows for arbitrary random 2-way delay distributions, does not rely on idealized instantaneous ACK, and thus would be more suitable for practical applications where random delay is present in both directions.

(iii) We derive the jointly optimal sensor/controller policy under the proposed new setting. The double relaxation in this work from a single-node policy to a joint policy and from 1-way delay to 2-way delay represents a significant advancement over the state of the art. Note that the existing works that focus on a single-node design [11], [12] only need to optimize *a single decision.* In contrast, in a two-node setting, the sensor-side optimal policy must take into account the optimal controller-side decision, which results in *two concatenated (and dependent) dynamic programming operations* that are jointly formulated and solved in this work for the first time in the literature.

(iv) The new unified framework includes many existing results as special cases, and we have used it to derive a new, optimal remote estimation scheme with 2-way delay, a strict generalization of [12]. (The optimal 2-way-delay AoI minimization results can be found in [18].)

The rest of the paper is organized as follows. In Sec. II, our detailed system model and problem formulation are presented. Our main results are outlined in Secs. III and IV. Sec. V uses the proposed framework to solve the remote estimation problem with random two-way delay. Numerical results are reported in Sec. VI. We conclude our work in Sec. VII. A subset of critical proofs will be provided in the appendix and the complete proofs can be found in the technical report [19].

II. Model and Formulation

A. System Model

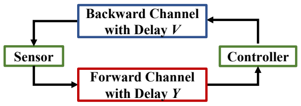

Backward Channel with Delay V
Sensor
Controller
Forward Channel with Delay Y

Fig. 1: A sensor/controller system with 2-way delay.

Our system model is best depicted in Fig. 1, which consists of a sensor, a controller, a forward sensor-to-controller channel and a backward controller-to-sensor channel. It is worth noting that we use the terms of sensor and controller in their broadest sense. The sensor node is not limited to a physical sensor that measures the location/temperature of the environment. Instead, it can be any data-generating node, e.g., a database server, a video-streaming source, etc. Also, the controller is not restricted to a node directly commanding an actuator. Instead, it can be any decision making component, e.g., computation of the inferred status of the remote database, or the video processing applications that render the actual video.

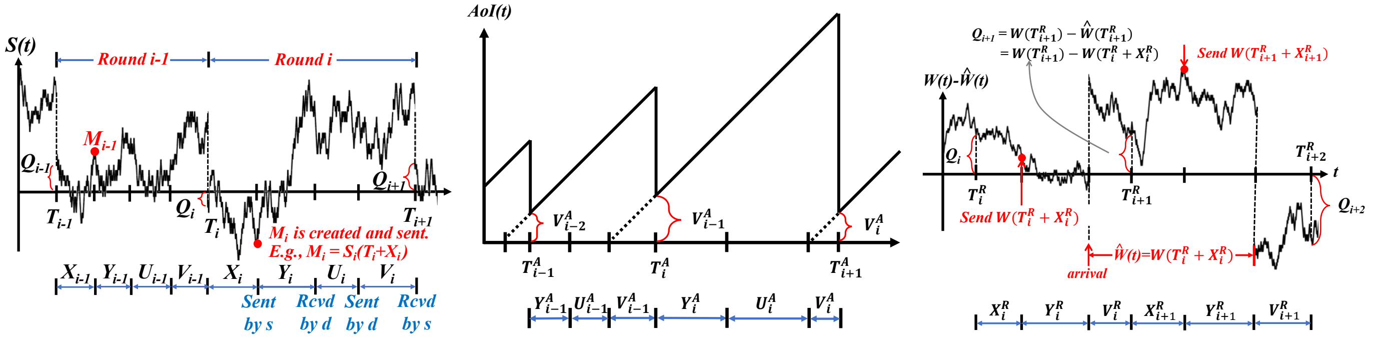
Each of the two channels incurs random transmission delay. With two-way delay in the communication loop, the information at the sensor and the controller is inherently delayed. Specifically, the waiting policy of the sensor (resp. controller) does not have instantaneous access to the status of the underlying network and has to wait for the delayed response from the controller (resp. sensor). *This two-way delay model and the resulting asymmetrically delayed information access where neither the sensor nor the controller has the instantaneous timing information is the most distinguishing feature of this work.* For comparison, most existing works [6]–[8], [11], [12] assume one node has instantaneous network-wide control information, which may not hold in practice where random delay is universally present.

We now explain our system model. We denote the system state as $S(t)$, for which we shift/relabel the values so that the origin $S(t) = 0$ is the most desired system state. The value of $S(t)$ may drift away from zero as time proceeds. We assume the evolution of $S(t)$ is related to a Wiener process $W(t)$ [20], a widely used (though idealized) model of the system state.[2] The detailed system evolution is defined as follows, and the corresponding illustration is provided in Fig. 2a.

Time sequences: The system consists of four discrete-time real-valued random processes X_i, Y_i, U_i, and V_i for all i. X_i is the i-th waiting time at the sensor; Y_i is the random delay for the i-th use of the sensor-to-controller channel; U_i is the i-th waiting time at the controller; V_i is the random delay for the i-th use of the controller-to-sensor channel.

The values of $\{X_i\}$ to $\{V_i\}$ can be used to derive another time sequence $\{T_i\}$ as follows: $T_1 \triangleq 0$ and $T_{i+1} \triangleq T_i + X_i + Y_i + U_i + V_i$ for all i. We call the interval $[T_i, T_{i+1})$ as the *i-th round,* which consists of the i-th waiting time of

[2] Some applications of the Wiener process model include unmanned aerial vehicles (UAVs) and mobile networks [21], [22].


S(t)
Round i-1
Round i
M_{i-1}
Q_{i-1}
Q_i
T_{i-1}
T_i
T_{i+1}
Q_{i+1}
M_i is created and sent.
E.g., $M_i = S_i(T_i+X_i)$
X_{i-1} Y_{i-1} U_{i-1} V_{i-1} X_i Y_i U_i V_i
Sent by s
Rcvd by d
Sent by d
Rcvd by s
AoI(t)
V^A_{i-2}
V^A_{i-1}
V^A_i
T^A_{i-1}
T^A_i
T^A_{i+1}
Y^A_{i-1} U^A_{i-1} V^A_{i-1} Y^A_i U^A_i V^A_i
$Q_{i+1} = W(T^R_{i+1}) - \hat{W}(T^R_{i+1})$
$= W(T^R_{i+1}) - W(T^R_i + X^R_i)$
Send $W(T^R_{i+1} + X^R_{i+1})$
$W(t)$-$\hat{W}(t)$
Q_i
T^R_i
Send $W(T^R_i + X^R_i)$
T^R_{i+1}
T^R_{i+2}
t
Q_{i+2}
arrival
$\hat{W}(t)=W(T^R_i + X^R_i)$
X^R_i Y^R_i V^R_i X^R_{i+1} Y^R_{i+1} V^R_{i+1}

(a) Our setting.

(b) Evolution of AoI.

(c) Evolution of $W(t) - \hat{W}(t)$ for remote estimation.

Fig. 2: Illustration of system evolution for different problem formulations.

the sensor, the delay of the i-th use of the forward channel, the i-th waiting time of the controller, and the delay of the i-th use of the backward channel. Clearly, the value T_i is the beginning of the i-th round.

We now describe the system behavior in the i-th round.

Reset-to-Q_i at the sensor: At time T_i, the sensor has received the message from the controller in the previous $(i-1)$-th round. It is very convenient to view the message as a *reset command.* We assume that upon receiving the reset command, the system state at time T_i will be reset to a random value Q_i, which is the (random) *initial value* of the i-th round. For example, with thermal noise it may be impossible to set the state value to be exactly 0 with infinite precision. The value Q_i thus models the residual randomness after reset, if any. We assume $\{Q_i\}$ is i.i.d. with $\mathbb{E}\{Q_i\} = 0$.

Remark 1: Again the term *reset* is used in the broadest sense. For example, in terms of data freshness control, *reset* could simply mean that the system state is changed from "stale" back to "fresh", not necessarily referring to a physical reset operation.

After reset-to-Q_i, the system state will evolve according to a Wiener process $W(t)$, until it is once again reset to Q_{i+1} at time T_{i+1}. The state value in the i-th round, denoted by $S_i(t)$, is thus described by

$$S_i(t) = W(t) - W(T_i) + Q_i, \text{ for } t \in [T_i, T_{i+1}). \quad (1)$$

We sometimes drop the subscript i and simply use $S(t)$.

Waiting time at the sensor: The sensor has the ability of waiting for an arbitrary amount of time $X_i \geq 0$, also see [9], [11]. The random variable X_i is a *stopping time* with respect to the filtration generated by $\{S(\tau) : \tau \leq t\}$ and the past acknowledgment packets. That is, the sensor observes the evolution of the system state and causally decides when to stop waiting and start transmission.

Upon transmission, the sensor sends (T_i, X_i, M_i) to the controller, where (T_i, X_i), defined in the previous paragraphs, serves as the time stamp(s) while M_i is the additional message/payload generated based on the past system states.

Random delay in the forward direction: The tuple (T_i, X_i, M_i) sent by the sensor at time $T_i + X_i$ will arrive at the controller at time $T_i + X_i + Y_i$. The transmission delay Y_i is i.i.d. and is independent from the rest of the system.

Waiting time at the controller: Since the message is time-stamped (containing (T_i, X_i)), the controller can infer the value of the forward transmission delay Y_i by subtracting $T_i + X_i$ from the actual arrival time $T_i + X_i + Y_i$. The waiting time $U_i \geq 0$ at the controller is then a function of all the previous messages and timing information $\{(T_j, X_j, Y_j, M_j) : j \leq i\}$.

Random delay in the backward direction: At time $T_i + X_i + Y_i + U_i$, the controller sends a reset signal, which will reach the sensor at time $T_{i+1} \triangleq T_i + X_i + Y_i + U_i + V_i$. The $(i+1)$-th round then begins, and we go back to reset-to-Q_{i+1} at the sensor. Again, we assume the backward delay V_i is i.i.d. and is independent from the rest of the system.

Technical assumptions: Similar to [11], [12], we assume (i) the statistics of $\{Q_i\}$, $\{Y_i\}$, and $\{V_i\}$ are known to both the sensor and the controller and $0 < \mathbb{E}\{Y_i\} + \mathbb{E}\{V_i\} < \infty$, and $\text{Var}\{Q_i\} + \text{Var}\{Y_i\} + \text{Var}\{V_i\} < \infty$; and (ii) the waiting times X_i and U_i must satisfy $\mathbb{E}\{X_i\} + \mathbb{E}\{U_i\} < \infty$. Any scheme with the waiting time $\mathbb{E}\{X_i\} = \infty$ or $\mathbb{E}\{U_i\} = \infty$ is deemed uninteresting because of its impact of infinitely delaying the next transmission in the average sense.

Remark 2: The non-negativity $X_i \geq 0$ (resp. $U_i \geq 0$) prohibits the sensor (resp. controller) from transmission before receiving the reset command (resp. message packet) from the controller (resp. sensor). This complies with the spirit of the stop-and-wait AQR mechanisms where the transmitter sends a new packet *after* receiving the ACK. In practice, ARQ is commonly used in wireless systems, e.g., IEEE 802.11e [23]. It is possible to design an even better scheme that transmits anticipatively before receiving any ACK, which, however, is beyond the scope of this work.

B. The Objective

For any given scheme $\{X_i\}$ and $\{U_i\}$, we define the *cost-aware L2 norm* (CAL2N) in the i-th round as

$$\mathbb{E}\left\{\int_{T_i}^{T_{i+1}} |S_i(t)|^2 \, dt\right\} + c_0 \quad (2)$$

where $S_i(t)$ is defined in (1) and we use its L2 norm to characterize how far it has drifted away from 0. The constant $c_0 \geq 0$ is the *cost of reset* in the end of the round. The value of

c_0 is chosen by the system designer and can be set to $c_0 = 0$ if desired.

Our goal is to minimize the *long-term average* CAL2N defined as follows.

$$\beta^*_{\text{CAL2N}} \triangleq \min_{\{X_i,M_i,U_i\}} \limsup_{n\to\infty} \frac{\sum\limits_{i=1}^{n} \left(\mathbb{E}\left\{\int_{T_i}^{T_{i+1}} |S_i(t)|^2\,dt\right\} + c_0\right)}{\sum\limits_{i=1}^{n} \Big(\mathbb{E}\left\{T_{i+1}-T_i\right\}+c_1\Big)} \tag{3}$$

where the non-negative constant $c_1 \geq 0$ in the denominator serves as a knob that determines whether we are biased towards a *duration-based* or *round-based* averaging. When $c_1 = 0$, the denominator is the total duration and (3) becomes the time-averaged CAL2N. For sufficiently large c_1, the denominator is approximately $c_1 n$ and (3) is proportional to the CAL2N averaged over n rounds, with each round having equal weight regardless how long/short it is. The value of c_1 is chosen by the system designer and can be set to 0 if desired.

To simplify (3), we notice that the optimization problem is a Markov decision problem with i.i.d.[3] Q_i, Y_i and V_i. As a result, it is sufficient to first find the optimal policy for the *single-round* optimization problem, assuming both the sensor and controller have access to some common randomness.[4] We can then apply the optimal single-round solution to every round. Following this reasoning, the equivalent single-round optimization problem becomes

$$\beta^*_{\text{CAL2N}} = \min_{(X,M,U)} \frac{\mathbb{E}\left\{\int_0^{X+Y+U+V} |S(t)|^2\,dt\right\} + c_0}{\mathbb{E}\left\{X+Y+U+V\right\}+c_1} \tag{4}$$

subject to X is a stopping time w.r.t. the filtration generated

$$\text{by } \{S(\tau), \tau \leq t\}; \tag{5}$$

$$\text{Payload } M \text{ is generated together with } X; \tag{6}$$

$$U \text{ is a function of } (M,X,Y); \tag{7}$$

$$\mathbb{E}\{X\} + \mathbb{E}\{U\} < \infty \tag{8}$$

where (5) to (8) follow from our discussion in Sec. II-A, and we drop the subscript i for notational simplicity.

If we are restricted to the dummy message $M = 0$, then the decision is made based on the *age* of the message (rather than the content itself). It is then similar to the AoI minimization problem (see Sec. II-C). On the other hand, if $M \neq 0$, then the content of the message could help decision making. In this case it is similar to the remote estimation problem (see Sec. II-D).

C. AoI Minimization Setting with Random Two-way Delay

Our setting can be viewed as a strict generalization of the AoI minimization problem [11] with random two-way delay described as follows. In the AoI minimization problem, the source sends packets to the destination through a queue that is not collocated with the source. We use T_i^A to denote the time instant at which the queue becomes empty for the i-th time. At time T_i^A, a notification packet will be sent from the queue (or equivalently from the destination) back to the source, which takes Y_i^A time to arrive. After receiving the notification, the source imposes a waiting time $U_i^A \geq 0$ and after that injects a new packet to the queue, which takes V_i^A time to be serviced. Once it is serviced, the queue becomes empty again (the $(i+1)$-th time). We thus have $T_{i+1}^A = T_i^A + Y_i^A + U_i^A + V_i^A$ and the process starts over. If we assume instantaneous feedback ($Y_i^A = 0$ with probability 1), the above problem formulation is identical to the basic system model of [11].

Suppose each packet is time-stamped and the AoI is defined as the current time minus the time stamp of the latest received packet [4]. Then the AoI grows linearly over time and is intermittently reset to V_i^A at time $T_{i+1}^A = T_i^A + Y_i^A + U_i^A + V_i^A$ since the time stamp of the latest arrival packet is $T_i^A + Y_i^A + U_i^A$. See [11] for more details and see Fig. 2b for illustration. The goal is to minimize the long-term average AoI, which is shown in [11] to be equivalent to solving the following single-round optimization problem:

$$\beta^*_A \triangleq \min_{U_i^A} \frac{\mathbb{E}\left\{\int_{V_{i-1}^A}^{V_{i-1}^A+Y_i^A+U_i^A+V_i^A} t\,dt\right\}}{\mathbb{E}\left\{Y_i^A+U_i^A+V_i^A\right\}}. \tag{9}$$

We now show that (9) is a special case of our setting defined in Secs. II-A and II-B. Note that the optimal U_i^A for the *one-way delay* setting has been solved in [11]. For the two-way delay setting (9), we have

Lemma 1: The optimal U_i^A for the AoI minimization problem (9) can be found by solving the optimal U_i^* in (4) of our setting while (i) assuming $c_0 = c_1 = 0$, (ii) hardwiring $X_i = M_i = 0$, i.e., forgoing the possibility of designing better X_i and M_i, (iii) setting the delays to be $Y_i = Y_i^A$, $V_i = V_i^A$, and (iv) choosing $Q_i = W(T_i) - W(T_i - V_{i-1})$.

Proof: Define the filtration until time $T_i + Y_i$ as $\mathcal{F}_i \triangleq \{(T_j, Y_j), : j \leq i\}$. We then have for any $t \in [T_i, T_{i+1})$,

$$\mathbb{E}\left\{|S_i(t)|^2 \middle| \mathcal{F}_i\right\} = \mathbb{E}\left\{|W(t) - W(T_i - V_{i-1})|^2 \middle| \mathcal{F}_i\right\} \tag{10}$$

$$= t - (T_i - V_{i-1}) \tag{11}$$

where (10) follows from (1) and our choice of Q_i in (iv); and (11) follows from the strong Markov property of the Wiener process. We can then rewrite the numerator of (4) as

$$\mathbb{E}\left\{\int_{T_i}^{T_i+Y_i+U_i+V_i} \mathbb{E}\left\{|S_i(t)|^2 \middle| \mathcal{F}_i\right\} dt\right\} \tag{12}$$

$$= \mathbb{E}\left\{\int_{T_i}^{T_i+Y_i+U_i+V_i} (V_{i-1} + t - T_i) dt\right\} \tag{13}$$

$$= \mathbb{E}\left\{\int_{V_{i-1}}^{V_{i-1}+Y_i+U_i+V_i} t dt\right\} \tag{14}$$

where (12) follows from Wald's lemma and the three facts (i) Y_i is deterministic once conditioning on $\mathcal{F}_i$; (ii) In Sec. II-A, the waiting time U_i at the controller is defined as a function

[3] More precise requirements are: (i) $\{Y_i\}$ and $\{V_i\}$ are i.i.d. and independent from the rest of the system; (ii) For any i, Q_i is independent of the waiting times $\{X_j, U_j : j < i\}$ in the previous rounds.

[4] The common randomness enables us to convert the temporal average over many rounds to the probabilistic average over a single round.

of $\{(T_j, X_j, Y_j, M_j) : j \leq i\}$. Since we set $X_i = M_i = 0$, it is clear that U_i is also deterministic once conditioning on $\mathcal{F}_i$; and (iii) V_i is independent of $\mathcal{F}_i$ and $|S_i(t)|^2$. Eq. (13) follows from (11); and (14) follows from the change of variables. This proves that our objective function (4) collapses to (9) once we set c_0, c_1, Y_i, V_i, X_i, M_i, U_i and Q_i properly. The high-level intuition is that the controller cannot directly observe the system state and has to make its decisions based on the *expected* cost $\mathbb{E}\{|S(t)|^2\}$, which has a linear growth rate with respect to the elapsed time (AoI) after the last reset. The CAL2N minimization problem thus includes the AoI minimization [11] as a special case. ■

D. Remote Estimation Setting with Random Two-way Delay

Next, we show that our setting is also a strict generalization of the remote estimation problem [12] with random two-way delay. The setting of [12] considers a system in which a sampler sends packets to an estimator through a queue that is not collocated with the sampler. Whenever the queue becomes empty, the queue sends a notification packet back to the sampler. We use T_i^R to denote the time instant at which the i-th notification packet has arrived at the sampler. After time T_i^R, the sampler continuously monitors an external random process $W(T_i^R + t)$, which is assumed to be a Wiener process. After some waiting time $X_i^R \geq 0$, which is a stopping time of $W(T_i^R + t)$, the sampler injects the latest observed value $W(T_i^R + X_i^R)$ to the queue, which takes Y_i^R time to be serviced. Once it is serviced, the queue becomes empty and the estimator receives the latest observation. Then a new notification packet is sent back to the sampler, which experiences some random delay V_i^R. Once the sampler receives the new $(i+1)$-th notification after delay V_i^R, the process starts over. It is clear that we have $T_{i+1}^R = T_i^R + X_i^R + Y_i^R + V_i^R$ in this system model.

We now describe the *estimation error* of this remote estimation system. Specifically, at time $t = T_i^R + X_i^R + Y_i^R$ the estimator receives the latest observed value $W(T_i^R + X_i^R)$ and uses it as an estimate of the external process $\hat{W}(t) = W(T_i^R + X_i^R)$ until the arrival of the next update packet at time $t = T_{i+1}^R + X_{i+1}^R + Y_{i+1}^R$. As a result, the estimation error $W(t) - \hat{W}(t)$ jumps to a new (smaller) initial value $W(T_i^R + X_i^R + Y_i^R) - W(T_i^R + X_i^R)$ at time $t = T_i^R + X_i^R + Y_i^R$. See Fig. 2c for illustration. Otherwise it evolves as a Wiener process until the arrival of the next observation $W(T_{i+1}^R + X_{i+1}^R)$ at time $t = T_{i+1}^R + X_{i+1}^R + Y_{i+1}^R$.

If we again use the single-round problem formulation, the optimization problem becomes:

$$\beta^*_{\text{MMSE}} \triangleq \min_{X_i^R} \frac{\mathbb{E}\left\{\int_{T_i^R}^{T_i^R+X_i^R+Y_i^R+V_i^R} \left(W(t) - \hat{W}(t)\right)^2 dt\right\}}{\mathbb{E}\left\{X_i^R + Y_i^R + V_i^R\right\}}. \quad (15)$$

Since $\hat{W}(t) = W(T_{i-1}^R + X_{i-1}^R)$ if $t \in [T_{i-1}^R + X_{i-1}^R + Y_{i-1}^R, T_i^R + X_i^R + Y_i^R]$, we can rewrite the numerator of (15) as

$$\mathbb{E}\left\{\int_{T_i^R}^{T_i^R+X_i^R+Y_i^R} \left(W(t) - W(T_{i-1}^R + X_{i-1}^R)\right)^2 dt\right\} + \mathbb{E}\left\{\int_0^{V_i^R} (W(Y_i^R + t) - W(0))^2 dt\right\} \quad (16)$$

where (16) uses the strong Markov property of the Wiener process and the assumption that (Y_i^R, V_i^R) is independent from the Wiener process.

We notice that the latter half of (16) can be further simplified as follows. Given $V_i^R = v$ and $Y_i^R = y$, we have

$$\mathbb{E}\left\{\int_0^{V_i^R} (W(Y_i^R + t) - W(0))^2 dt \Big| V_i^R = v, Y_i^R = y\right\}$$
$$= \int_0^v \mathbb{E}\left\{(W(y+t) - W(0))^2\right\} dt \quad (17)$$
$$= \int_0^v (y+t) dt = yv + \frac{v^2}{2} \quad (18)$$

where (17) follows from the fact that V_i^R and Y_i^R are independent from the rest of the system; and (18) follows from the strong Markov property of the Wiener process.

By further taking the expectation of (18) over the i.i.d. V_i^R and Y_i^R, the numerator of (15) can be rewritten as

$$\mathbb{E}\left\{\int_{T_i^R}^{T_i^R+X_i^R+Y_i^R} \left(W(t) - W(T_{i-1}^R + X_{i-1}^R)\right)^2 dt\right\} + \mathbb{E}\{Y_i^R\}\mathbb{E}\{V_i^R\} + \frac{\mathbb{E}\{(V_i^R)^2\}}{2}. \quad (19)$$

We now show that the above remote estimation problem (15) is a special case of our setting in Secs. II-A and II-B. Note that the optimal X_i^R for the *one-way delay* setting has been solved in [12]. For the two-way delay setting (15), we have

Lemma 2: The optimal X_i^R for the remote estimation problem (15) can be found by solving the optimal X_i^* in (4) of our setting while (i) hardwiring $M_i = U_i = V_i = 0$, i.e., short-circuiting the controller and the backward delay and using a dummy message $M_i = 0$, (ii) setting the forward delay to be $Y_i = Y_i^A$, (iii) choosing $c_0 = \mathbb{E}\{Y_i^R\}\mathbb{E}\{V_i^R\} + \frac{\mathbb{E}\{(V_i^R)^2\}}{2}$, $c_1 = \mathbb{E}\{V_i^R\}$, and (iv) $Q_i = W(T_i) - W(T_{i-1} + X_{i-1})$.

Proof: Note that by (1) and the choices of $U_i = V_i = 0$, we can rewrite the numerator of (4) as

$$\mathbb{E}\left\{\int_{T_i}^{T_i+X_i+Y_i} (W(t) - W(T_i) + Q_i)^2 dt\right\} + c_0. \quad (20)$$

Then by the special choices of c_0 and Q_i in (iii) and (iv), it is straightforward to verify that the numerator of (4) (i.e., (20)) is identical to the numerator of (15) (i.e., (16)). It is also straightforward to verify that the denominators of (15) and (4) are identical. Our objective function (4) thus collapses to the one for remote estimation with two-way delay (15) once we set c_0, c_1, Y_i, V_i, X_i, M_i, U_i and Q_i properly. The high-level intuition is that having direct observation of the system state $S(t)$, the sensor naturally has the same role as the *sampler* in the context of remote estimation of a Wiener process [12]. ■

Remark 3: [12] considered remote estimation with *one-way* delay, which is a special case of the one in this subsection by further setting controller-to-sensor delay $V_i^R = 0$ (and hence $c_0 = c_1 = 0$).

Remark 4: In some sense, the two important papers [11], [12] form a perfect pair. The former AoI minimization work focuses on the controller action (assuming $X_i = Y_i = 0$) without using the state information, while the latter remote-estimation work studies the sensor action (assuming $U_i = V_i = 0$) that directly observes the state $S(t)$. A main contribution of this work is to unify these two results and study the optimal sensor/controller scheme that jointly optimizes X_i and U_i.

III. Main Results — The Optimal Policies

In this section, we will present three policies: (i) the jointly optimal policy (X^*, M^*, U^*), (ii) the optimal No-Wait-At-Sensor (NWAS) policy which imposes $X_i = 0$ and optimizes the rest of the system, and (iii) the optimal No-Wait-At-Controller (NWAC) policy which imposes $U_i = 0$ and optimizes the rest. Policies (ii) and (iii) are meant for scenarios in which either the sensor or the controller is forced to adopt a suboptimal zero-wait policy due to other system-level considerations.

A. An Auxiliary Minimization Problem

Given the distributions of the i.i.d. $\{Y_i\}$, $\{V_i\}$, $\{Q_i\}$ and any constant values $c_0, c_1 \geq 0$, for any $\beta \in (-\infty, \infty)$ we define $p(\beta)$ as the optimal value of the following minimization problem:

$$p(\beta) \triangleq \inf_{(X,M,U)} \mathbb{E}\left\{\int_0^{X+Y+U+V} |S(t)|^2 \, dt\right\} + c_0 - \beta\left(\mathbb{E}\{X+Y+U+V\} + c_1\right) \quad (21)$$

subject to (5) to (8)

where we drop the subscript i for simplicity. We then have

Proposition 1: (i) The function $p(\beta)$ is concave, continuous, and strictly decreasing, (ii) there exists a unique $\beta^* \in [0, \beta_{\text{max}}]$ such that $p(\beta^*) = 0$, where

$$\beta_{\text{max}} \triangleq \frac{\mathbb{E}\{Y\}\mathbb{E}\{V\} + \frac{1}{2}\mathbb{E}\left\{Y^2 + V^2\right\} + c_0}{\mathbb{E}\{Y+V\} + c_1} + \frac{\mathbb{E}\{Q^2\} \cdot \mathbb{E}\{Y+V\}}{\mathbb{E}\{Y+V\} + c_1} \quad (22)$$

(iii) the unique solution β^* is identical to the β^*_{CAL2N} defined in (4), and (iv) The (X, M, U) scheme that attains $p(\beta^*)$ also achieves the β^*_{CAL2N} in (4).

Proof: See Appendix A of the technical report [19]. ■

By Proposition 1, the optimization problem (4) can be solved using the following steps: For any given β, we first find the optimal (X, M, U) that minimizes (21) and the corresponding $p(\beta)$ value. We then find the optimal $\beta^* = \beta^*_{\text{CAL2N}}$ by a bisection search over $[0, \beta_{\text{max}}]$. In the sequel, we discuss how to find the optimal (X, M, U) solution of $p(\beta)$ in (21) for any given β.

B. Optimal Waiting Time at the Controller

Define $\overline{M}_i = (T_i, X_i, M_i)$. The following proposition holds for any arbitrary message scheme $\{M_i\}$.

Proposition 2: Given any arbitrary payload $\{M_i\}$ and any $\beta > 0$, the optimal waiting time $U^*_{i|M}$ at the controller that minimizes (21) is as follows.

$$U^*_{i|M} = \max\left(\beta - \left(Y_i + \mathbb{E}\left\{(S_i(T_i + X_i))^2 \,\middle|\, \overline{M}^{(i)}\right\} + \mathbb{E}\{V_i\}\right), 0\right) \quad (23)$$

where $\overline{M}^{(i)} \triangleq \left\{\overline{M}_j : j \leq i\right\}$. That is, the optimal controller is a *water-filling policy* that calculates the difference between β and $\left(Y_i + \mathbb{E}\{(S_i(T_i + X_i))^2 \,|\, \overline{M}^{(i)}\} + \mathbb{E}\{V_i\}\right)$. See Appendix B of the technical report [19] for the proof.

Corollary 1: If the dummy message $M_i = 0$ is used, then for any given $\beta > 0$, the optimal waiting time $U^*_{i|M=0}$ at the controller that minimizes (21) becomes

$$U^*_{i|M=0} = \max\left(\beta - \left(Y_i + \left(\mathbb{E}\{Q_i^2\} + X_i\right) + \mathbb{E}\{V_i\}\right), 0\right). \quad (24)$$

A high-level interpretation of (24) is as follows. If the dummy message $M_i = 0$ is used, the controller can only estimate the system state $\mathbb{E}\{(S_i(T_i + X_i))^2 \,|\, \overline{M}^{(i)}\}$ by the timing information (X_i, Y_i) and the knowledge that $S_i(t)$ is reset to Q_i at time $t = T_i$. Eq. (24) then follows from (1), (23) and the properties of the Wiener process $S_i(t)$. See Appendix O of the technical report [19] for a detailed proof.

From (24), two levels of randomness are observed: (i) X_i could be optimally/suboptimally chosen; (ii) Even when X_i is optimally chosen, different random realizations of Y_i may still lead to different waiting time $U^*_{i|M=0}$.

C. Optimal Message Sent by the Sensor

Proposition 3: The optimal message that minimizes (21) is $M_i^* = S_i(T_i + X_i)$, the latest state value at the time of transmission $T_i + X_i$.

This result follows directly from the fact that the system state is a strong Markov process and thus the latest system state consists of all the information the controller can possibly need. We omit the proof due to the space limit. Combining Propositions 2 and 3, we immediately have

Corollary 2: With the optimal message M_i^* in Proposition 3, the optimal waiting time at the controller becomes

$$U_i^* \triangleq U^*_{i|M^*} = \max\left(\beta - \left(Y_i + (S_i(T_i + X_i))^2 + \mathbb{E}\{V_i\}\right), 0\right). \quad (25)$$

D. Optimal Waiting Time at the Sensor

The design of the sensor waiting time X has to take into account the sent message M and the controller waiting time U. In the sequel we exclusively assume M_i^* in Proposition 3 is used. Two different controller schemes $U = U_i^*$ and $U = 0$ are considered. When U_i^* is used, we denote the corresponding optimal sensor scheme by X_i^*, i.e., the tuple (X_i^*, M_i^*, U_i^*) represents the jointly optimal sensor/controller policy. When $U = 0$ is used, we denote the corresponding optimal sensor scheme by $X_{i|\text{ZW}}^*$, which represents the best-possible X if the controller employs a zero-wait (ZW) policy. The tuple $(X_{i|\text{ZW}}^*, M_i^*, U_i = 0)$ is thus what we previously referred to as the optimal No-Wait-At-Controller (NWAC) policy. Note that the optimal No-Wait-At-Sensor (NWAS) policy is the combination of $(X_i = 0, M_i^*, U_i^*)$, which has been fully described in Proposition 3 and Corollary 2 and is thus not the focus of this subsection.

We first describe how to find jointly optimal X_i^*, and later present how the procedure of finding X_i^* can be modified to find $X_{i|\text{ZW}}^*$. For any $s \in (-\infty, \infty)$, define two functions $g_\beta(s)$ and $h_\beta(s)$ by

$$g_\beta(s) \triangleq a_{s,4} \cdot s^4 + a_{s,2} \cdot s^2 + a_{s,0} + a_0 \tag{26}$$

$$h_\beta(s) \triangleq g_\beta(s) - (\beta s^2 - \frac{1}{6} s^4) \tag{27}$$

where

$$a_{s,4} \triangleq -\frac{\mathbb{P}\left(s^2 + Y \leq \beta - \mathbb{E}\{V\}\right)}{2} \tag{28}$$

$$\begin{aligned} a_{s,2} \triangleq\ & \mathbb{E}\{Y + V\} \\ & + \mathbb{E}\left\{\mathbb{1}_{\{s^2 + Y \leq \beta - \mathbb{E}\{V\}\}} \cdot (\beta - \mathbb{E}\{V\} - Y)\right\} \end{aligned} \tag{29}$$

$$a_{s,0} \triangleq -\frac{\mathbb{E}\left\{\mathbb{1}_{\{s^2 + Y \leq \beta - \mathbb{E}\{V\}\}} \cdot (\beta - \mathbb{E}\{V\} - Y)^2\right\}}{2} \tag{30}$$

$$\begin{aligned} a_0 \triangleq\ & -\beta\left(\mathbb{E}\{Y + V\} + c_1\right) + \mathbb{E}\{Y\}\mathbb{E}\{V\} \\ & + \frac{1}{2}\mathbb{E}\left\{Y^2 + V^2\right\} + c_0 \end{aligned} \tag{31}$$

and $\mathbb{1}_{\{\cdot\}}$ is the indicator function. Note that $g_\beta(s)$ and $h_\beta(s)$ are generally not fourth-order polynomials since the coefficients $a_{s,4}$, $a_{s,2}$, and $a_{s,0}$ also depend on s.

Lemma 3: For any $\beta \geq 0$ and any distributions of Y and V, which can be discrete, continuous, or hybrid, both functions $g_\beta(s)$ and $h_\beta(s)$ are even and continuous. Furthermore, $h_\beta(s)$ is lower bounded by a shifted 4-th order polynomial of s, i.e., $h_\beta(s) \geq \frac{1}{12}s^4 + (a_0 - 3\beta^2)$ for all $s \in (-\infty, \infty)$.

For any two functions f_1 and f_2, we say $f_1 \prec f_2$ if $f_1(s) \leq f_2(s), \forall s \in (-\infty, \infty)$. The *convex hull* (also called *the lower convex envelope*) of the function $h_\beta(s)$ is defined as

$$\text{Cnvx}\big(h_\beta(s)\big) \triangleq \sup\{f(s) : f \text{ is convex}, f \prec h_\beta\}. \tag{32}$$

Corollary 3: For any $\beta \geq 0$, the lower convex envelope $\text{Cnvx}\big(h_\beta(s)\big)$ is finite for all $s \in (-\infty, \infty)$.

Corollary 3 follows directly from Lemma 3. The proofs of Lemma 3 and Corollary 3 are relegated to Appendix D of the technical report [19].

We now describe the optimal sensor waiting time X_i^*.

Proposition 4: For any given β, the optimal X_i^* that minimizes (21) is the hitting time:

$$X_i^* = \inf\{t \geq 0 : S_i(T_i + t) \in \mathcal{S}_{\text{tx},\beta}\} \tag{33}$$

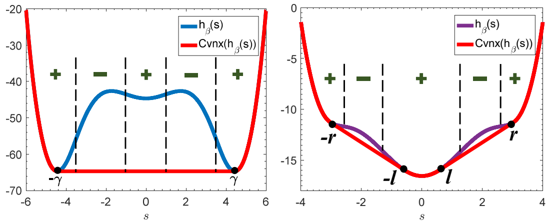
where the set $\mathcal{S}_{\text{tx},\beta}$, called *the transmission set*, is the collection of all state values s satisfying

$$\text{Cnvx}\left(h_\beta(s)\right) = h_\beta(s) \tag{34}$$

i.e., the set of s whose corresponding values of the convex hull function are equal to those of the original function $h_\beta(s)$.

Proof: This is the most involved proof of this work and we provide a detailed sketch of the proof in Appendix A. In Appendix A, we also discuss the general ideas of the proof of Proposition 2. See Appendix E of the technical report [19] for complete proof of Proposition 4. ■

Fig. 3a illustrates a (piecewise) even function $h_\beta(s)$, which contains 5 pieces with the corresponding second-order derivatives being $+-+-+$ if we scan the s values from $-\infty$ to ∞. Fig. 3a also plots the convex hull function $\text{Cnvx}\left(h_\beta(s)\right)$. One can see that in this example, $\text{Cnvx}\left(h_\beta(s)\right) = h_\beta(s)$ iff $|s| \geq \gamma$ for some threshold γ. As a result, $\mathcal{S}_{\text{tx},\beta} = \{s : |s| \geq \gamma\}$. The optimal X_i^* is thus the first time when $|S_i(T_i + t)|$ hits γ.



(a) (b)

Fig. 3: Examples of $h_\beta(s)$ and $\text{Cnvx}(h_\beta(s))$.

Another example of $h_\beta(s)$ is plotted in Fig. 3b, for which $\text{Cnvx}\left(h_\beta(s)\right) = h_\beta(s)$ iff s belongs to neither of the two (symmetric) intervals $(-r, -l)$ and (l, r). In this example, the sensor transmits if either $|S(T_i + t)| \geq r$ or $|S(T_i + t)| \leq l$.

Proposition 4 describes the X_i^* in the jointly optimal sensor/controller scheme. In the following we elaborate how we derive $X_{i|\text{ZW}}^*$ for the optimal NWAC policy.

Define a new $g_{\text{NWAC},\beta}(s)$ by

$$\begin{aligned} g_{\text{NWAC},\beta}(s) \triangleq & \mathbb{E}\{Y + V\}\, s^2 + \mathbb{E}\{Y\}\mathbb{E}\{V\} + \frac{1}{2}\mathbb{E}\left\{Y^2 + V^2\right\} \\ & - \beta\left(\mathbb{E}\{Y + V\} + c_1\right) + c_0. \end{aligned} \tag{35}$$

Note that the $g_{\text{NWAC},\beta}(s)$ is a second-order polynomial of s since its coefficients do not depend on s.

By substituting $g_\beta(s) = g_{\text{NWAC},\beta}(s)$ in (26) and repeating the steps listed (27), (32), and Proposition 4, we can find the optimal waiting time $X_{i|\text{ZW}}^*$ of the best NWAC policy. Specifically, for any given $\beta \in (-\infty, \infty)$, define

$$h_{\text{NWAC},\beta}(s) \triangleq g_{\text{NWAC},\beta}(s) - (\beta s^2 - \frac{1}{6}s^4). \tag{36}$$

Since $g_{\text{NWAC},\beta}(s)$ has a nice form of being a second-order polynomial, by simple calculus one can verify that

$$\text{Cnvx}(h_{\text{NWAC},\beta}(s)) = \begin{cases} h_{\text{NWAC},\beta}(s) & \text{if } s^2 \geq \gamma_{\text{NWAC}} \\ h_{\text{NWAC},\beta}(\sqrt{\gamma_{\text{NWAC}}}) & \text{if } s^2 < \gamma_{\text{NWAC}} \end{cases} \tag{37}$$

$$\text{where} \quad \gamma_{\text{NWAC}} \triangleq \max(3 \cdot (\beta - \mathbb{E}\{Y + V\}), 0) \tag{38}$$

is a constant threshold.

Proposition 5: Using the definition of γ_{NWAC} in (38), the optimal waiting time $X^*_{i|\text{ZW}}$ is the hitting time:

$$X^*_{i|\text{ZW}} = \inf\{t \geq 0 : |S_i(T_i + t)|^2 \geq \gamma_{\text{NWAC}}\}. \tag{39}$$

Proof: See Appendix H of the technical report [19] . ■

Intuitively, the difference between X^*_i and $X^*_{i|\text{ZW}}$ is due to different schemes used at the controller, and the sensor thus has to react differently. Propositions 4 and 5 prove that the effects of different controller schemes can be summarized either as the function $g_\beta(s)$ in (26) or $g_{\text{NWAC},\beta}(s)$ in (35). The actual optimization mechanisms at the sensor remain the same and are described by the steps of finding $h_\beta(s)$ and the convex hull $\text{Cnvx}\big(h_\beta(s)\big)$, and comparing $h_\beta(s)$ and $\text{Cnvx}\big(h_\beta(s)\big)$ to decide the corresponding transmission set $\mathcal{S}_{\text{tx},\beta}$.

E. Finding the Optimal β^*

All the previous discussions assume an arbitrarily given β. We now describe how to find β^*. We first discuss the case for the jointly optimal policy, and then describe the cases for the optimal NWAC and NWAS policies.

Proposition 1 shows that the optimal β^* for the jointly optimal scheme (X^*_i, M^*_i, U^*_i) is the unique root of $p(\beta) = 0$ defined in (21). We now describe how to find $p(\beta)$. Recall that Q_i describes the random state value in the beginning of the i-th round (right after reset). We then have

Proposition 6: For any $\beta \geq 0$, define $\phi(\beta, s)$ as

$$\phi(\beta, s) \triangleq \text{Cnvx}\left(h_\beta(s)\right) + (\beta s^2 - \frac{1}{6}s^4) \tag{40}$$

where $h_\beta(s)$ was first defined in (26) to (31). The optimal $p(\beta)$ in (21) can be computed by

$$p(\beta) = \mathbb{E}_Q\{\phi(\beta, Q)\} \tag{41}$$

i.e., we first assume the s value in $\phi(\beta, s)$ is randomly distributed with distribution Q and then evaluate $p(\beta)$ by finding the expectation in (41). See Appendix E of the technical report [19] for the proof.

Proposition 7: For any $\beta \geq 0$, define $\phi_{\text{NWAC}}(\beta, s)$ and $p_{\text{NWAC}}(\beta)$ as

$$\phi_{\text{NWAC}}(\beta, s) \triangleq \text{Cnvx}(h_{\text{NWAC},\beta}(s)) + (\beta s^2 - \frac{1}{6}s^4) \tag{42}$$

$$p_{\text{NWAC}}(\beta) \triangleq \mathbb{E}_Q\{\phi_{\text{NWAC}}(\beta, Q)\} \tag{43}$$

where $h_{\text{NWAC},\beta}(s)$ was first defined in (35) and (36). The optimal β^* for the optimal NWAC scheme (i.e., $(X^*_{i|\text{ZW}}, M^*_i, U_i = 0)$) is the unique root of $p_{\text{NWAC}}(\beta) = 0$. See Appendix H of the technical report [19] for the proof.

Proposition 8: For any $\beta \geq 0$, define $\phi_{\text{NWAS}}(\beta, s)$ and $p_{\text{NWAS}}(\beta)$ as

$$\phi_{\text{NWAS}}(\beta, s) \triangleq g_\beta\,(s) \tag{44}$$

$$p_{\text{NWAS}}(\beta) \triangleq \mathbb{E}_Q\{\phi_{\text{NWAS}}(\beta, Q)\} \tag{45}$$

where g_β was first defined in (26). The optimal β^* for the optimal NWAS scheme (i.e., $(X_i = 0, M^*_i, U^*_i)$) is the unique root of $p_{\text{NWAS}}(\beta) = 0$. See Appendix I of the technical report [19] for the proof.

Comparing Propositions 6 and 8, we notice that both propositions are very similar in the sense that they first find a function $\phi(\beta, s)$ and then evaluate the corresponding $p(\beta)$ by taking the expectation over Q. The differences are that the $\phi(\beta, s)$ in Proposition 6 is obtained by applying a sequence of convex-hull-based operations to $g_\beta(s)$ in (26), (27), (32), and (40), whereas Proposition 8 directly sets $\phi_{\text{NWAS}}(\beta, s) = g_\beta(s)$. The intuition is that in the NWAS policy, the sensor always chooses $X_i = 0$ without any optimization/minimization. Therefore the initial function $g_\beta(s)$, which is the objective function based on the optimal controller waiting time U^*_i, will be directly used as the $\phi_{\text{NWAS}}(\beta, s)$. This essentially skips the intermediate optimization/minimization steps in (26), (27), (32), and (40) that compute $\phi(\beta, s)$ in (40) from $g_\beta(s)$, which captures the effects of using optimal X^*_i. Once we swap (40) with (44), the steps of (41) and (45) are identical.

We also compare Propositions 6 and 7. Since $g_\beta(s)$ represents the effect of the optimal controller waiting time U^*_i, when shifting from optimal U^*_i to zero $U_i = 0$ in NWAC, the only change is to replace $g_\beta(s)$ in (26) by the $g_{\text{NWAC},\beta}(s)$ in (35). The remaining steps, i.e., {(27), (40), (41)} versus {(36), (42), (43)}, are identical.

F. Complexity of Finding the Jointly Optimal Scheme

We first summarize the detailed steps of finding the jointly optimal sensor/controller policy and the corresponding β^*.

Step 1: For any β, compute the functions $g_\beta(s)$, $h_\beta(s)$, and $\phi(\beta, s)$ by (26), (27), and (40), and then compute the value of $p(\beta)$ by (41).

Step 2: Repeatedly use Step 1 and the bisection search over $\beta \in [0, \beta_{\max}]$ to find the unique β^* satisfying $p(\beta^*) = 0$.

Step 3: Substitute $\beta = \beta^*$ in Secs. III-B and III-D to derive the respective optimal policies for the controller and the sensor.

We note that the bisection steps, i.e., Steps 2 and 3, also appear in [11], [12] and thus do not incur additional complexity. For some special delay distributions Y_i and V_i and the reset distribution Q_i, e.g., exponential, it is possible to derive closed-form expressions of $g_\beta(\cdot)$, $h_\beta(\cdot)$, $\phi(\beta, \cdot)$, and $p(\beta)$ by calculus. For arbitrary Y_i, V_i, and Q_i distributions, we can compute $g_\beta(\cdot)$ and $h_\beta(\cdot)$ by quantizing the continuous s values into discrete points. Then we can use existing *linear-time* algorithms, e.g., [24], to compute the convex hull $\text{Cnvx}(h_\beta(s))$. The expectation step in (41) can subsequently be computed in linear time as well. Overall, the complexity of our algorithm is identical to [11], [12]; that is, are all linear-time in terms of the number of quantization points.

IV. Further Examination of The Optimal Policy

In this section, we prove some properties of the jointly optimal sensor/controller scheme (X^*_i, M^*_i, U^*_i).

Lemma 4: $\mathcal{S}_{\text{tx},\beta^*}$ is symmetric over $s = 0$, i.e., for any $s \in (-\infty, \infty)$, $s \in \mathcal{S}_{\text{tx},\beta^*}$ if and only if $(-s) \in \mathcal{S}_{\text{tx},\beta^*}$.

Proof: This lemma follows directly from Lemma 3 and the definition of $\mathcal{S}_{\text{tx},\beta}$ in Proposition 4. ■

Define $\mathcal{S}^c_{\text{tx},\beta^*} \triangleq (-\infty, \infty)\backslash\mathcal{S}_{\text{tx},\beta^*}$ as the complement of $\mathcal{S}_{\text{tx},\beta^*}$. We then have the following self-explanatory lemma.

Lemma 5: $\mathcal{S}^{c}_{\text{tx},\beta^*}$ must be a collection of disjoint open intervals (l_i, r_i), namely,

$$\mathcal{S}^{c}_{\text{tx},\beta^*} = \bigcup_{i=1}^{\mu}(l_i, r_i) \tag{46}$$

where μ is the total number of disjoint open intervals, and $\{(l_i, r_i) : i\}$ satisfies $-\infty < l_i < r_i < \infty$ for all $i \in [1, \mu]$.

Lemmas 4 and 5 imply that if $\mu = 1$, then $\mathcal{S}_{\text{tx},\beta^*} = \{s : |s| \geq \gamma\}$ for some $\gamma > 0$, which is termed *the threshold policy* in [12]. Similarly, if $\mu = 0$, then $\mathcal{S}_{\text{tx},\beta^*} = (-\infty, \infty)$ and the optimal policy is a zero-wait policy. In the sequel, we examine the value of μ, calculated by (46), for various scenarios.

A. Deterministic Forward Transmission Delay $Y_i = y_0$

Proposition 9: If there exists a constant y_0 such that $\mathbb{P}(Y_i = y_0) = 1$, then we always have $\mu \leq 1$ and $\mathbb{P}(U_i^* = 0) = 1$.

Proof: See Appendix J of the technical report [19]. ∎

In other words, with deterministic forward transmission delay Y_i, the optimal sensor policy is either a zero-wait policy ($\mu = 0$) or a threshold policy ($\mu = 1$), and the optimal controller strategy is always a zero-wait policy regardless of the distribution of backward delay V_i.

B. Exponential Forward Transmission Delay Y_i

Proposition 10: If $Y_i \sim \text{Exp}(\lambda_Y)$ is exponentially distributed with service rate $\lambda_Y > 0$, then we always have $\mu \leq 2$.

Proof: See Appendix K of the technical report [19]. ∎

If we choose $Y \sim \text{Exp}(\lambda_Y)$ with $\lambda_Y = 0.2$, $V \sim \text{Exp}(\lambda_V)$ with $\lambda_V = 6$, $c_0 = 20$, $c_1 = 0$, and $Q \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = 0.125$, then we can numerically compute $\beta^* = 7.236$ using the 3 steps in Sec. III-E. The resulting $\mathcal{S}_{\text{tx},\beta^*} = \{s : |s| \leq 0.012 \text{ or } 0.082 \leq |s|\}$ indeed has $\mu = 2$. The upper bound $\mu \leq 2$ in Proposition 10 is thus tight.

We call the $\mu = 2$ policy an *interval policy*. The reason is that with $\mu = 2$ the transmission set $\mathcal{S}_{\text{tx},\beta^*}$ is of the form

$$\mathcal{S}_{\text{tx},\beta^*} = \{s : |s| \leq l \text{ or } r \leq |s|\} \tag{47}$$

for a pair of $0 < l < r < \infty$. That is, the optimal sensor scheme should transmit when the system state is either too large $|s| \geq r$ or too small $|s| \leq l$. At the first glance, this strategy seems counterintuitive due to the following reason: Our goal is to minimize the average value of $|S_i(t)|^2$. Therefore, large $|s|$ is considered to be "bad" and small $|s|$ is considered to be "good". An intuitive strategy inspired by [11] and [12] is to hold off transmission (i.e., to wait) when the state is good (when $|s|$ is small) in order to prolong the duration of staying in a good state. Our results show that under the setting of joint sensor/controller coordination, the sensor sometimes should transmit when the state becomes *too good* (when $|s| \leq l$).

One explanation of this surprising phenomenon is as follows. The goal of CAL2N minimization in (4) is for the sensor and the controller to jointly design their strategies and *one thus has to decide how to split the waiting time between the sensor and the controller.* A deeper look shows that each of them has its unique advantages and disadvantages. In particular, the sensor is able to observe the full system state $S_i(t)$ continuously and use it to make its decision X_i. The controller cannot observe $S_i(t)$ directly, but instead it can directly observe the realization of the random sensor-to-controller delay Y_i, a valuable piece of information known exclusively to the controller. Therefore, when the system state is very good, $|s|$ being small, there is a bigger chance that the controller will see a good expected system state[5] $Y_i + s^2 + \mathbb{E}\{V_i\}$ in (23). As a result, the sensor should transmit so that the controller, which has the additional observation of Y_i, can make a better informed decision U_i^* to further extend the duration of staying in a good system state. *One of the contributions of this work is to uncover this unexpected sensor/controller coordination that is critical to achieving the optimal performance.*

The above discussion also explains the intuition of Proposition 9. With deterministic Y_i, the controller has a strictly inferior set of information since the observed Y_i is a constant. Hence, all the waiting time should be allocated to the sensor, i.e., $\mathbb{P}(U_i^* = 0) = 1$, and the sensor transmits if and only if the system state is bad (either a zero-wait or a threshold policy $|s| \geq \gamma$), which corresponds to $\mu \leq 1$.

C. A Special Case of $\mu = 6$

The coordination between the sensor and the controller can sometimes be very subtle and beyond the high-level intuition discussed previously. Consider the following example.

Example 1: Consider the distribution of Y being

$$\begin{cases} \mathbb{P}(Y = 6) = 0.35, & \mathbb{P}(Y = 45) = 0.06, \\ \mathbb{P}(Y = 51) = 0.08, & \mathbb{P}(Y = 53) = 0.08, \\ \mathbb{P}(Y = 54) = 0.23, & \mathbb{P}(Y = 90) = 0.2, \end{cases} \tag{48}$$

$\mathbb{P}(V = 20) = 1$, $c_0 = 45$, $c_1 = 0$, and the initial random variable $Q \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = 6.6$. We can numerically compute $\beta^* = 80.049$ using the 3 steps in Sec. III-E, and the resulting $\mathcal{S}_{\text{tx},\beta^*}$ is

$$\begin{aligned} \mathcal{S}_{\text{tx},\beta^*} = \{s : |s| \leq 1.803 \text{ or } 3.100 \leq |s| \leq 3.795 \\ \text{or } 3.858 \leq |s| \leq 6.767 \text{ or } 7.305 \leq |s|\} \end{aligned} \tag{49}$$

which has $\mu = 6$.

The reason of having a highly fractured transmission set $\mathcal{S}_{\text{tx},\beta^*}$ is due to the delicate probabilistic balance between the benefits of observing $S_i(t)$ at the sensor and observing Y_i at the controller.

V. Solving Two-way Remote Estimation Problem

In this section, we derive[6] the optimal Wiener-process remote estimation scheme with random delay Y_i^R and V_i^R in the forward and feedback directions, a generalization of the results in [12].

In particular, by Lemma 2 we can find the optimal remote estimation solution by finding the best NWAC scheme $X^*_{i|\text{ZW}}$ (under the constraint $U_i = 0$) in Propositions 5 and 7.

[5] Since the optimal $M_i^* = S_i(T_i + X_i)$ is used, the term $\mathbb{E}\{(S_i(T_i + X_i))^2|\overline{M}^{(i)}\} = (S_i(T_i + X_i))^2 = s^2$ is directly related to the value of $|s|$.

[6] One can use similar techniques to solve AoI minimization problem with random two-way delay. Due to space limits, we refer the readers to [18] for the final results.

Specifically, plugging the parameter values in Lemma 2 into (35), we have

$$g_{\text{NWAC},\beta}(s) \triangleq \mathbb{E}\left\{Y^R\right\}s^2 + \frac{1}{2}\mathbb{E}\left\{(Y^R)^2\right\}$$
$$- \beta\left(\mathbb{E}\{Y^R\} + \mathbb{E}\{V^R\}\right) + \mathbb{E}\{Y^R\}\mathbb{E}\{V^R\} + \frac{\mathbb{E}\{(V^R)^2\}}{2}.$$

By Proposition 5, the optimal transmission policy $X^*_{i|\text{ZW}}$ for the sensor is to transmit whenever $s^2 \geq \gamma^*_{\text{NWAC}}$, where

$$\gamma^*_{\text{NWAC}} = \max(3\left(\beta^*_{\text{NWAC}} - \mathbb{E}\{Y^R\}\right), 0) \tag{50}$$

and the β^*_{NWAC} used in (50) can be computed by finding the root of the $p_{\text{NWAC}}(\beta)$ defined in Proposition 7.

In the following, we show the above description of how to compute the optimal threshold γ^*_{NWAC} can be further simplified to the following equivalent form.

Lemma 6: The γ^*_{NWAC} defined in (50) and Proposition 7 can also be computed by finding the root of

$$\frac{\mathbb{E}\left\{\max(\gamma^2, (W(Y^R))^4)\right\} + 3\mathbb{E}\{(V^R)^2\}}{2\gamma}$$
$$= \mathbb{E}\left\{\max\left(\gamma, (W(Y^R))^2\right)\right\} + \mathbb{E}\{V^R\}. \tag{51}$$

See Appendix L of the technical report [19] for the proof.

Corollary 4: If we further limit ourselves to the 1-way delay setting, i.e., $V^R_i = 0$, then the optimal $X^*_{i|\text{ZW}}$ policy described in (50) and Lemma 6 reproduces the optimal 1-way-delay remote-estimation scheme in [12, Theorem 1].

Proof: By specializing (51) in Lemma 6 with the one-way delay setting $V^R_i = 0$, the optimal remote estimation scheme in (50) has the corresponding γ^*_{NWAC} value being a root of

$$\frac{\mathbb{E}\left\{\max(\gamma^2, (W(Y^R))^4)\right\}}{2\gamma} = \mathbb{E}\left\{\max(\gamma, (W(Y^R))^2)\right\} \tag{52}$$

which is identical to the Eq. (15) in [12]. Hence, the optimal scheme in (50) and Lemma 6 reproduces [12, Theorem 1]. ■

VI. Simulation Results

We compare the performance of our jointly optimal sensor/controller policy and five other important alternatives.

(i) Zero-wait (ZW) $(X_i = 0, U_i = 0)$ [10]: The zero-wait policy is commonly known as the work-conserving policy [10].

(ii) Optimal No-Wait-At-Sensor (NWAS) policy $(X_i = 0, M^*_i, U^*_i)$: See the discussion in Secs. III-B and III-C.

(iii) Optimal No-Wait-At-Controller (NWAC) policy $(X^*_{i|\text{ZW}}, M^*_i, U_i = 0)$: See Sec. III-D.

(iv) AoI-minimization scheme (AoI-min) [11]: It is related to the NWAS $(X_i = 0)$ scheme. The differences are (i) It does not take into account the reset cost c_0 and the per-round cost c_1, see (3); (ii) It falsely assumes the forward delay $Y_i = 0$ even though the actual Y_i could be non-zero; (iii) It employs the suboptimal message $M_i = 0$ instead of the optimal $M^*_i = S_i(T_i + X_i)$; (iv) It assumes $Q_i = W(T_i) - W(T_i - V_{i-1})$ regardless of the actual Q_i; and (v) It hardwires $X_i = 0$ and optimizes the U_i under the suboptimal assumptions (i)–(iv). We are interested in measuring the performance loss (compared to the optimal NWAS scheme) due to these suboptimal decisions.

(v) Remote-estimation scheme (RE) [12]: As discussed in Sec. II-D, it is an instance of NWAC schemes. The differences between the RE and the optimal NWAC schemes are (i) RE does not take into account the reset cost c_0 and the per-round cost c_1 in (3); and (ii) RE falsely assumes the backward delay $V_i = 0$ even though the actual V_i could be non-zero; and (iii) RE assumes $Q_i = W(T_i) - W(T_{i-1} + X_{i-1})$ irrespective of the actual Q_i.

We report the results for exponential forward and backward delays, while similar behaviors can be observed for log-normal delays. The initial value Q_i is assumed to be Gaussian with zero mean and variance σ^2. The results are presented in Fig. 4.

In Fig. 4a we notice that the larger the σ value, the wider the range of the initial value Q_i, which models the case of less accurate reset/control. Hence, the CAL2N of all 5 schemes increases as σ goes up.

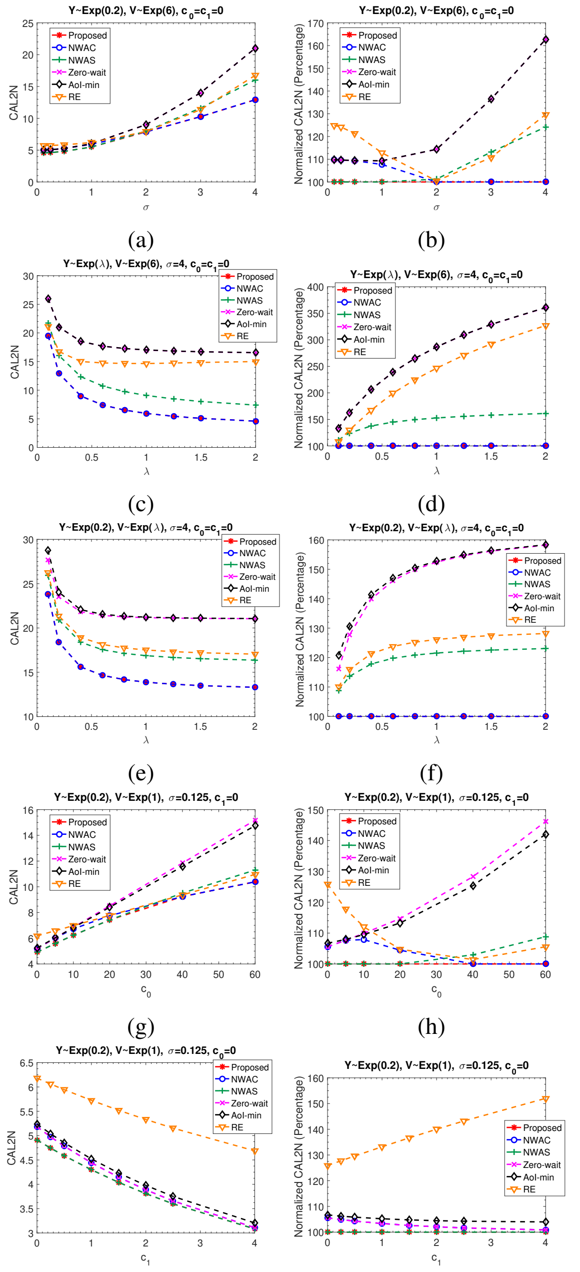
A more interesting comparison is to calculate the ratio of the CAL2N of any scheme over that of our scheme, i.e., the normalized CAL2N plotted in Fig. 4b. Indeed, the normalized CAL2N of any scheme is always $\geq 100\%$ since our scheme is provably optimal.

In Fig. 4b we also observe that when the reset is accurate (small σ), the performance of the optimal NWAS is identical to that given by the optimal solution, which implies the jointly optimal scheme will allocate all its waiting time to the controller and perform zero-wait at the sensor. On the other hand, when the reset is loose (large σ), the jointly optimal scheme will allocate all its waiting time to the sensor and perform zero-wait at the controller, i.e., the optimal NWAC becomes globally optimal for large σ. In either case, our algorithm optimally splits the waiting time between the sensor and the controller and always attains the best performance.

As shown in Figs. 4c and 4e, we fix $\sigma = 4$ and vary the delay distribution parameters λ_Y and λ_V, respectively. In both figures, similar trends can be observed: When either λ_Y or λ_V increases (namely, when the expected delay is shorter), the CAL2N of any scheme goes down. Interestingly, the performance of the optimal NWAC is as good as the jointly optimal solution in both cases. It appears that in these scenarios, the reset quality σ value, see Figs. 4a and 4b, has stronger impact on whether NWAC is jointly optimal or not than the delay distributions of Y and V.

In Figs. 4g and 4i, we consider different values of c_0 and c_1. As can be seen, the optimal split of the waiting time between the sensor and the controller is heavily dependent on the value of reset cost c_0, see Figs. 4g and 4h, but much less on the per-round cost c_1, see Figs. 4i and 4j. Overall, from Figs. 4a to 4j one can see that each of the five alternatives excels in some scenarios but performs poorly in the others, while our scheme always achieves the optimal performance.

We are particularly interested in the relative performance of the existing 1-way-delay-based AoI-min and RE schemes when there is 2-way delay in the system. Because existing results do not take into account 2-way delay, as expected, the AoI-min scheme (resp. RE scheme) is always worse than the optimal NWAS scheme (resp. NWAC scheme) and is much worse than the jointly optimal solution. Furthermore, considering only 1-way delay (i.e., the AoI-min and RE schemes) and ignoring the delay in the other direction can be quite detrimental. In many cases they perform worse than the naive zero-wait solution. See RE vs. ZW in Fig. 4b and AoI-min vs. ZW in Fig. 4f. In particular, RE is significantly worse the ZW in Fig. 4b, while the difference between AoI-min and ZW in Fig. 4f is much smaller.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Fig. 4: Long-term average CAL2N for various settings. Those on the left are on absolute scale and those on the right are normalized with respect to the CAL2N of the optimal policy.

VII. Conclusion

We have proposed a new Wiener-process-based framework and characterized the corresponding jointly optimal sensor/controller policy, which unifies AoI minimization and remote estimation, two recent important results that have spawned substantial interests in the literature. The consideration of the two-way delay model and joint sensor-&-controller design have successfully addressed the asymmetrically delayed information access of the practical systems and represents a significant improvement over the existing results based on idealized zero-delay acknowledgment feedback.

Appendix A
Proof of Proposition 4

We present sketches of the proof of Proposition 4, which consist of the following components: **Component 1:** The joint time and space quantization and the corresponding dynamic programming (DP) problem after quantization; If we call the new DP problem as *quantized-DP*, **Component 2** describes how the controller-side policy affects the initialization of the sensor-side quantized-DP solver; Using Components 1 and 2, one can numerically solve the quantized-DP problem assuming a finite-horizon $\overline{J} < \infty$ setting. However, the brute-force DP iterations often obscure the physical meaning/interpretation of the optimal decision rules and do not provide any closed-form solution. To uncover further results, **Component 3** focuses on solving quantized-DP analytically through careful convergence analysis when $\overline{J} \to \infty$. Finally, **Component 4** discusses how (i) the initialization, (ii) the closed-form optimal decision rule, and (iii) the closed-form optimal objective values can be seamlessly converted back to the corresponding parts in the original continuous time/space problem.

A. *Component 1: Joint time and space quantization*

Our optimization has a continuous time domain and a continuous state-space domain and the goal is to solve

$$f^{[2]}(s) = \min_{X} \mathbb{E}\left\{ \int_0^{X+Y+U^*+V} |S(t)|^2\, dt \,\middle|\, S(0) = s \right\} + c_0 \\ - \beta \left(\mathbb{E}\left\{ X + Y + U^* + V \,\middle|\, S(0) = s \right\} + c_1 \right). \quad (53)$$

We now describe how to quantize the above continuous problem to its discrete version.

We first consider a discrete-time binary symmetric random walk $\mathcal{B}(i)$ such that the initial point $\mathcal{B}(0) \in \mathbb{Z}$ can be arbitrarily chosen and $\mathcal{B}(i+1) = \mathcal{B}(i) + (1 - 2b(i))$, $i \geq 0$, where $\{b(i) \in \{0,1\} : i\}$ are i.i.d. Bernoulli random variables with $p = 0.5$.[7] We can use the discrete-time $\mathcal{B}(i)$ to construct a continuous-time process $W^{\delta}(t) = \sqrt{\delta} \cdot \mathcal{B}\left(\left\lfloor \frac{t}{\delta} \right\rfloor\right)$. It is well known that $W^{\delta}(t)$ converges to a Wiener process[7] $W(t)$ in distribution when $\delta \to 0$ [20]. In the quantized setting, we thus assume the system state $S(t)$ evolves according to $W^{\delta}(t)$.

For the time-domain quantization, we quantize the time-domain variables with quantization step δ, namely defining $\overline{X} = \left\lfloor \frac{X}{\delta} \right\rfloor$, $\overline{Y} = \left\lfloor \frac{Y}{\delta} \right\rfloor$, $\overline{U} = \left\lfloor \frac{U}{\delta} \right\rfloor$, $\overline{V} = \left\lfloor \frac{V}{\delta} \right\rfloor$, and $\overline{\beta} = \left\lfloor \frac{\beta}{\delta} \right\rfloor$ as the integer approximations of X, Y, U, V, β. For the state-space quantization, we quantize the state-space values with quantization step size $\sqrt{\delta}$. Given $S(0)$, we define the quantized

[7] In the literature [20], a random walk and a Wiener process typically starts with $\mathcal{B}(0) = 0$ and $W(0) = 0$. Here we relax this constraint to allow $\mathcal{B}(0)$ and $W(0)$ to be of any value, which is notationally convenient since we allow for the Wiener process that is periodically "reset" to Q_i.

state $i = \left\lfloor \frac{S(0)}{\sqrt{\delta}} \right\rfloor$. Also, define $\overline{c}_0 = \lfloor \frac{c_0}{\delta^2} \rfloor$, and $\overline{c}_1 = \lfloor \frac{c_1}{\delta} \rfloor$. See the technical report [19] for details.

Using the above integer approximations, we consider the finite-horizon (sensor-side) discrete optimization problem

$$\begin{aligned} & F^{[2]}(i, \overline{J}) \\ & = \min_{\overline{X} \le \overline{J}} \mathbb{E} \left\{ \sum_{k=1}^{\overline{X}+\overline{Y}+\overline{U}^*+\overline{V}} \left(i^2 + (\mathcal{B}(k) - \mathcal{B}(0))^2 \right) \middle| \mathcal{B}(0) = i \right\} \\ & - \overline{\beta} \left(\mathbb{E} \left\{ \overline{X} + \overline{Y} + \overline{U}^* + \overline{V} \middle| \mathcal{B}(0) = i \right\} + \overline{c}_1 \right) + \overline{c}_0 \qquad (54) \end{aligned}$$

where $\overline{X} \le \overline{J}$ is the finite horizon constraint. The above can then be mapped to the original continuous problem by

Lemma 7:

$$f^{[2]}(s) = \lim_{\delta \to 0} \lim_{\overline{J} \to \infty} \delta^2 F^{[2]} \left(\left\lfloor \frac{s}{\sqrt{\delta}} \right\rfloor, \overline{J} \right). \qquad (55)$$

While the proof of this lemma is quite technical (see Appendix M of the technical report [19] for details), the intuition is straightforward. That is, letting $\overline{J} \to \infty$ enables us to find the optimal solution of quantized-DP in a non-constrained, infinite horizon setting. Then taking $\delta \to 0$ converts the quantized-DP problem back to the original continuous time/space optimization problem in (53), which is of infinite horizon. The proof effectively contains two halves, with a greater focus on establishing the former, i.e., proving $F^{[2]}(i, \infty) = \lim_{\overline{J} \to \infty} F^{[2]}(i, \overline{J})$.

We solve $F^{[2]}(i, \overline{J})$ by dynamic programming over finite horizon $[0, \overline{J}]$. For convenience, we use the reverse time index j during the iterative Bellman equation, where $j = 0$ (resp. $j = \overline{J}$) represents the last slot (resp. the first slot). Define

$$\begin{aligned} & F^{[2]}(i, j) = \min_{\overline{X}} \\ & \mathbb{E} \Bigg\{ \sum_{k=\overline{J}-j+1}^{\overline{X}+\overline{Y}+\overline{U}^*+\overline{V}} \left(i^2 + (\mathcal{B}(k) - \mathcal{B}(\overline{J}-j))^2 \right) \Bigg| \mathcal{B}(\overline{J}-j) = i, \\ & \qquad \qquad \qquad \qquad \overline{X} \ge (\overline{J}-j) \Bigg\} \\ & - \overline{\beta} \left(\mathbb{E} \left\{ \overline{X} - (\overline{J}-j) + \overline{U}^* \middle| \mathcal{B}(\overline{J}-j) = i, \overline{X} \ge (\overline{J}-j) \right\} \right) \\ & - \overline{\beta} \left(\mathbb{E} \left\{ \overline{Y} + \overline{V} \right\} + \overline{c}_1 \right) + \overline{c}_0 \qquad (56) \\ & = \begin{cases} F^{[2]}_{\mathsf{Send}}(i, j) & \text{if } j = 0 \\ \min \left(F^{[2]}_{\mathsf{Send}}(i, j), F^{[2]}_{\mathsf{Wait}}(i, j) \right) & \text{if } 1 \le j \le \overline{J}. \end{cases} \qquad (57) \end{aligned}$$

It is self-explanatory to verify that the above definition of (56) is consistent with (54) once we set $j = \overline{J}$. Eq. (57) is the boundary condition of the dynamic programming.

B. Component 2: Initialization of the sensor-side quantized-DP solver

We now describe how the initialization of the above sensor-side problem (i.e., solving for optimal $\overline{X}^*$) is related to the optimal controller-side quantized-DP ($\overline{U}^*$ in Proposition 2). Details of the controller-side DP are provided in [19] and we provide a summary of some properties of $\overline{U}^*$ here. Since we use the optimal $M_i^* = S_i(T_i + X_i)$ in Proposition 3, the optimal U^* is a function[8] of $|S_i(T_i + X_i)|$ and Y_i. Hence, if we define the controller-side quantized state value i_1 by

$$i_1 = \left\lfloor \frac{|S(T_i + X_i)|}{\sqrt{\delta}} \right\rfloor \qquad (58)$$

and define $\overline{y} = \lfloor \frac{Y_i}{\delta} \rfloor$, then we can also convert the controller-side DP problem to its quantized version. Specifically,

$$\begin{aligned} & F^{[1]}(i_1, \overline{y}, \overline{J}) = \min_{\overline{U}(i_1, \overline{y}) \le \overline{J}} \\ & \mathbb{E} \left\{ \sum_{k=1}^{\overline{U}+\overline{V}} \left((i_1)^2 + \overline{y} + (\mathcal{B}(\overline{x} + \overline{y} + k) - \mathcal{B}(\overline{x} + \overline{y}))^2 \right) \right\} \\ & - \overline{\beta} \cdot \mathbb{E} \left\{ \overline{U} \right\} \qquad (59) \end{aligned}$$

where $\overline{U}$ is the quantized controller waiting time, and $\overline{J}$ is the finite-horizon constraint on $\overline{U} \le \overline{J}$ as well. See Appendix B of the technical report [19] for detailed discussion.

We now describe how $F^{[1]}$ is used when solving the sensor-side DP problem in (54) to (57). To that end, we first expand the objective function of $F^{[2]}_{\mathsf{Send}}$ in (57) as follows.

$$\begin{aligned} & F^{[2]}_{\mathsf{Send}}(i, j) \\ & = \mathbb{E} \Bigg\{ \sum_{k=\overline{J}-j+1}^{(\overline{J}-j)+\overline{Y}+\overline{U}^*+\overline{V}} \left(i^2 + (\mathcal{B}(k) - \mathcal{B}(\overline{J}-j))^2 \right) \\ & \qquad \qquad \qquad \Bigg| \mathcal{B}(\overline{J}-j) = i, \overline{X} = \overline{J}-j \Bigg\} \\ & - \overline{\beta} \Bigg(\mathbb{E} \Bigg\{ \overline{Y} + \overline{U}^* + \overline{V} \Bigg| \mathcal{B}(\overline{J}-j) = i, \overline{X} = \overline{J}-j \Bigg\} + \overline{c}_1 \Bigg) \\ & + \overline{c}_0 \qquad (60) \end{aligned}$$

which is obtained from (56) by hardwiring $\overline{X} = \overline{J} - j$. We can also write $F^{[2]}_{\mathsf{Wait}}(i, j)$ in (57) iteratively (since we use the *reverse time index*, the DP backward iteration will be carried out from $j - 1$ to j).

$$\begin{aligned} F^{[2]}_{\mathsf{Wait}}(i, j) = & \, (i)^2 - \overline{\beta} + 1 \\ & + 0.5 \left(F^{[2]}(i+1, j-1) + F^{[2]}(i-1, j-1) \right) \qquad (61) \end{aligned}$$

where $(i)^2 - \overline{\beta} + 1$ is the immediate cost and the rest is the expected future cost computed from the previous iteration. Since we consider a symmetric binary random walk, the probability that the discrete state i in next time slot increases (or decreases) by 1 is 0.5.

To establish the connection between $F^{[2]}_{\mathsf{Send}}$ and $F^{[1]}$, we notice that $F^{[2]}_{\mathsf{Send}}(i, j) = F^{[2]}_{\mathsf{Send,part1}}(i, j) + F^{[2]}_{\mathsf{Send,part2}}(i, j)$ where

$$\begin{aligned} F^{[2]}_{\mathsf{Send,part1}}(i, j) \triangleq \mathbb{E} \Bigg\{ & \sum_{k=\overline{J}-j+1}^{(\overline{J}-j)+\overline{Y}} \left(i^2 + (\mathcal{B}(k) - \mathcal{B}(\overline{J}-j))^2 \right) \Bigg\} \\ & - \overline{\beta} \Bigg(\mathbb{E} \Bigg\{ \overline{Y} + \overline{V} + \overline{c}_1 \Bigg) + \overline{c}_0 \qquad (62) \end{aligned}$$

[8] Here we use the symmetry between $S_i(\cdot) > 0$ versus $S_i(\cdot) < 0$.

and $F^{[2]}_{\mathsf{Send,part2}}(i,j)$

$$\triangleq \mathbb{E}\Bigg\{\sum_{k=\overline{J}-j+\overline{Y}+1}^{(\overline{J}-j)+\overline{Y}+\overline{U}^*+\overline{V}} \left(i^2+(\mathcal{B}(k)-\mathcal{B}(\overline{J}-j))^2\right) \Bigg| \mathcal{B}(\overline{J}-j)=i,\overline{X}=\overline{J}-j\Bigg\} - \overline{\beta}\Bigg(\mathbb{E}\Big\{\overline{U}^*\Big|\mathcal{B}(\overline{J}-j)=i,\overline{X}=\overline{J}-j\Big\}\Bigg). \tag{63}$$

That is, we split the summation in (60) into two pieces. The conditioning event inside $F^{[2]}_{\mathsf{Send,part1}}(i,j)$ is dropped due to the strong Markov property of $\mathcal{B}(k)$ and independence of the delays $\overline{Y}$ and $\overline{V}$ from the rest of the system.

By further using the strong Markov property of $\mathcal{B}(k)$, the term $F^{[2]}_{\mathsf{Send,part1}}(i,j)$ in (62) can be simplified to

$$F^{[2]}_{\mathsf{Send,part1}}(i,j) = \mathbb{E}\left\{\overline{Y}\right\}(i)^2+\frac{1}{2}\mathbb{E}\left\{(\overline{Y})^2\right\}+\frac{1}{2}\mathbb{E}\{\overline{Y}\} - \overline{\beta}\left(\mathbb{E}\{\overline{Y}+\overline{V}\}+\overline{c}_1\right)+\overline{c}_0. \tag{64}$$

By comparing (59) and (63), we also have

$$F^{[2]}_{\mathsf{Send,part2}}(i,j) = \mathbb{E}_{\overline{Y}}\left\{F^{[1]}(i,\overline{Y},\overline{J})\right\}. \tag{65}$$

Namely, $F^{[2]}_{\mathsf{Send,part2}}(i,j)$, the (partial) cost of Send at the sensor side, is the expectation of the controller-side cost $F^{[1]}(i,\overline{Y},\overline{J})$ averaged over different random realizations of the delay $\overline{Y}$.

Putting everything together, the $F^{[2]}_{\mathsf{Send}}(i,j)$ in (57) is related to the controller-side DP problem $F^{[1]}(i,\overline{Y},\overline{J})$ by

$$F^{[2]}_{\mathsf{Send}}(i,j) = \mathbb{E}\left\{\overline{Y}\right\}(i)^2+\frac{1}{2}\mathbb{E}\left\{(\overline{Y})^2\right\}+\frac{1}{2}\mathbb{E}\{\overline{Y}\} - \overline{\beta}\left(\mathbb{E}\{\overline{Y}+\overline{V}\}+\overline{c}_1\right)+\overline{c}_0+\mathbb{E}_{\overline{Y}}\left\{F^{[1]}(i,\overline{Y},\overline{J})\right\} \tag{66}$$

which, together with (61), completes the sensor-side quantized DP problem in (57). In the sequel we analyze the properties of the corresponding optimal solution.

C. Component 3: Analytically solving the quantized-DP

We first note that $F^{[2]}_{\mathsf{Send}}(i,j)$ in (66) does not depend on the value of j. The intuition behind is that the cost of Send from time $\overline{J}-j+1$ onward, see (60), only depends on the state value i but not on the time stamp $\overline{J}-j$. As a result, by the boundary condition in (57), we have

$$F^{[2]}_{\mathsf{Send}}(i,j) = F^{[2]}_{\mathsf{Send}}(i,0) = F^{[2]}(i,0). \tag{67}$$

By (57), (61) and (67), we have the following more compact form of iteration that directly operates on $F^{[2]}(i,j)$: $\forall j\geq 1$,

$$F^{[2]}(i,j) = \min\big(F^{[2]}(i,0),\ i^2-\overline{\beta}+1 + 0.5(F^{[2]}(i+1,j-1)+F^{[2]}(i-1,j-1))\big). \tag{68}$$

To analyze (68), we notice that the polynomial

$$A(i) \triangleq -\frac{i^4}{6}+(\overline{\beta}-\frac{5}{6})i^2 \tag{69}$$

satisfies the difference equation

$$A(i) = i^2-\overline{\beta}+1+0.5(A(i+1)+A(i-1)). \tag{70}$$

If we define another function

$$H^{[2]}(i,j) \triangleq F^{[2]}(i,j) - A(i) \tag{71}$$

then $H^{[2]}(i,j)$ must satisfy the following iterative equation

$$H^{[2]}(i,j) = \min\Big(H^{[2]}(i,0), 0.5\left(H^{[2]}(i+1,j-1)+H^{[2]}(i-1,j-1)\right)\Big) \tag{72}$$

due to (68) and (70). Namely, the new function $H^{[2]}(i,j)$ absorbs the $i^2-\overline{\beta}+1$ term in (68). Hence the new $H^{[2]}(i,j)$ follows a *homogeneous* iterative equation in (72) while the original $F^{[2]}(i,j)$ follows a non-homogeneous iterative equation in (68). Our approach is to first compute/evaluate $H^{[2]}(i,j)$ directly using the homogeneous iteration (72) and then compute[9] retrospectively $F^{[2]}(i,j) = H^{[2]}(i,j)+A(i)$.

In the following, we analyze $\lim_{\overline{J}\to\infty}H^{[2]}(i,\overline{J})$, i.e., the limiting results of the iterative computation in (72). The proofs for the following Lemmas 8 to 11 are provided in the technical report [19]. We first notice that

Lemma 8: By iteratively applying (72), we have

$$H^{[2]}(i,j+1) \leq H^{[2]}(i,j) \tag{73}$$

for any i and $j\geq 0$.

Definition 1: A discrete function $f(i):\mathbb{Z}\mapsto\mathbb{R}$ is said to be *d.convex* if

$$f(i) \leq 0.5\left(f(i+1)+f(i-1)\right), \forall i\in\mathbb{Z}. \tag{74}$$

Lemma 9: There exists a d.convex function $H_{\mathrm{LB}}(i)$ such that

$$-\infty < H_{\mathrm{LB}}(i) \leq H^{[2]}(i,j) \text{ for all } i \text{ and } j. \tag{75}$$

By Lemmas 8 and 9, and the Monotone Convergence Theorem, $\lim_{\overline{J}\to\infty}H^{[2]}(i,\overline{J})$ exists. We now derive $\lim_{\overline{J}\to\infty}H^{[2]}(i,\overline{J})$. We notice that $H^{[2]}(i,0)$, the value of $H^{[2]}(i,j)$ with j set to zero, is a function of i. We then define the *discrete convex hull*[10] of the function $H^{[2]}(i,0)$ as

$$\mathrm{D.Cnvx}(H^{[2]}(i,0)) \triangleq \sup\{f(i): f\prec H^{[2]}(i,0), f \text{ is d.convex}\}. \tag{76}$$

By Lemma 9, $\mathrm{D.Cnvx}(H^{[2]}(i,0))$, which is a function of i, always exists since the supremum is taken over a non-empty set. Furthermore, $\mathrm{D.Cnvx}(H^{[2]}(i,0))$ is also d.convex due to similar reasons that the (continuous) convex hull is itself convex. We now present the following lemmas:

Lemma 10: For any arbitrarily given integer i, we have

$$\lim_{\overline{J}\to\infty}H^{[2]}(i,\overline{J}) \leq \mathrm{D.Cnvx}(H^{[2]}(i,0)). \tag{77}$$

Lemma 11: $\mathrm{D.Cnvx}(H^{[2]}(i,0)) \leq \lim_{\overline{J}\to\infty}H^{[2]}(i,\overline{J})$.

Lemmas 10 and 11 jointly imply

$$\lim_{\overline{J}\to\infty}H^{[2]}(i,\overline{J}) = \mathrm{D.Cnvx}(H^{[2]}(i,0)). \tag{78}$$

Thus far, we have proven that the dynamic programming iterations in (57), (61), and (66) can be solved by (i) computing $H^{[2]}(i,0)\triangleq F^{[2]}(i,0)-A(i)$ in (71), (ii) finding the $\mathrm{D.Cnvx}(H^{[2]}(i,0))$, and (iii) using the equality $\lim_{\overline{J}\to\infty}F^{[2]}(i,\overline{J}) = \mathrm{D.Cnvx}(H^{[2]}(i,0))+A(i)$. The final

[9] This spirit is closely related to the free-boundary method, which, however, directly solves the optimization problem that has a continuous time domain and a continuous state-space domain. See, e.g., Eqs. (75) to (78) in [25].

[10] This convex hull definition is quite standard. Also see [26].

step of the proof is to convert the operations/steps of discrete-time, discrete-space solutions back to the original continuous-time, continuous-space solution.

D. Component 4: Converting back to the continuous time/space problem

In (66) and (67), we established the expression of $F^{[2]}(i,0)$, the boundary value of the sensor-side quantized finite-horizon DP problem where the finite horizon is denoted by $\overline{J}$. Obviously, different $\overline{J}$ will lead to different $F^{[2]}(i,0)$, see (67) and the appearance of $\overline{J}$ in (66). We then have

Lemma 12: Denote the continuous-time continuous-space state value as $S(t)=s$. We have

$$\lim_{\delta\to 0}\lim_{\overline{J}\to\infty}\delta^2 F^{[2]}\left(\left\lfloor\frac{s}{\sqrt{\delta}}\right\rfloor,0\right)=g_\beta(s) \tag{79}$$

which is defined in (26). See the technical note [19] for the proof.

Lemma 12 shows that $g_\beta(s)$ corresponds to the starting point (the last slot) of the dynamic programming solver. In the following we will show that in the continuous-time continuous-space domain, the iterative computation in (57), (61), and (66) corresponds to the computation of $h_\beta(s)$ and the convex hull operations defined in Proposition 4.

By (69), one can easily see that $\lim_{\delta\to 0}\delta^2 A\left(\left\lfloor\frac{s}{\sqrt{\delta}}\right\rfloor\right)=-\frac{s^4}{6}+\beta s^2$. Combining this observation with (71) and (79), we have

$$\lim_{\delta\to 0}\delta^2 H^{[2]}\left(\left\lfloor\frac{s}{\sqrt{\delta}}\right\rfloor,0\right)=h_\beta(s)=g_\beta(s)-(\beta s^2-\frac{1}{6}s^4).$$

Finally, by translating the discrete convex hull relationship in (78) to its continuous convex hull counterpart, we have

Lemma 13:

$$\begin{aligned}&\lim_{\delta\to 0}\lim_{\overline{J}\to\infty}\delta^2 H^{[2]}(i,\overline{J})=\mathrm{Cnvx}\left(g_\beta(s)-\left(\beta s^2-\frac{1}{6}s^4\right)\right)\\&=\mathrm{Cnvx}(h_\beta(s)).\end{aligned} \tag{80}$$

We now describe the optimal sensor policy. Define $H^{[2]}(i,\infty)\triangleq\lim_{\overline{J}\to\infty}H^{[2]}(i,\overline{J})$. Note that by (72) and comparing it to the original versions (57) and (68), the sensor should send if $H^{[2]}(i,\infty)=H^{[2]}(i,0)$ and should wait if

$$H^{[2]}(i,\infty)=0.5\left(H^{[2]}(i+1,\infty)+H^{[2]}(i-1,\infty)\right). \tag{81}$$

By translating the above discrete-time decision back to its continuous-time domain, the optimal sensor policy is to transmit if and only if $S(t)=s$ satisfying $\mathrm{Cnvx}(h_\beta(s))=h_\beta(s)$. The sketch of the proof of Proposition 4 is thus complete.

References

[1] L. M. Huyett, E. Dassau, H. C. Zisser, and F. J. Doyle III, "Glucose sensor dynamics and the artificial pancreas: The impact of lag on sensor measurement and controller performance," *IEEE Control Systems Magazine*, vol. 38, no. 1, pp. 30–46, 2018.

[2] A. Javed, H. Larijani, A. Ahmadinia, R. Emmanuel, M. Mannion, and D. Gibson, "Design and implementation of a cloud enabled random neural network-based decentralized smart controller with intelligent sensor nodes for HVAC," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 393–403, 2016.

[3] T. Li, K. Keahey, K. Wang, D. Zhao, and I. Raicu, "A dynamically scalable cloud data infrastructure for sensor networks," in *Proceedings of the 6th Workshop on Scientific Cloud Computing*. ACM, 2015, pp. 25–28.

[4] X. Song and J. W.-S. Liu, "Performance of multiversion concurrency control algorithms in maintaining temporal consistency," in *Proceedings., Fourteenth Annual International Computer Software and Applications Conference*. IEEE, 1990, pp. 132–139.

[5] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *2012 Proceedings IEEE INFOCOM*. IEEE, 2012, pp. 2731–2735.

[6] C. Kam, S. Kompella, and A. Ephremides, "Age of information under random updates," in *2013 IEEE International Symposium on Information Theory*. IEEE, 2013, pp. 66–70.

[7] C. Kam, S. Kompella, G. D. Nguyen, and A. Ephremides, "Effect of message transmission path diversity on status age," *IEEE Transactions on Information Theory*, vol. 62, no. 3, pp. 1360–1374, 2015.

[8] M. Costa, M. Codreanu, and A. Ephremides, "Age of information with packet management," in *2014 IEEE International Symposium on Information Theory*. IEEE, 2014, pp. 1583–1587.

[9] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," in *2015 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2015, pp. 3008–3012.

[10] L. Kleinrock, *Queueing systems, volume 2: Computer applications*. Wiley-Interscience, 1976.

[11] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," *IEEE Transactions on Information Theory*, vol. 63, no. 11, pp. 7492–7508, 2017.

[12] 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.

[13] K. J. Astrom and B. M. Bernhardsson, "Comparison of Riemann and Lebesgue sampling for first order stochastic systems," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 2. IEEE, 2002, pp. 2011–2016.

[14] B. Hajek, K. Mitzel, and S. Yang, "Paging and registration in cellular networks: Jointly optimal policies and an iterative algorithm," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, vol. 1. IEEE, 2003, pp. 524–532.

[15] X. Gao, E. Akyol, and T. Başar, "Optimal estimation with limited measurements and noisy communication," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 1775–1780.

[16] K. Nar and T. Başar, "Sampling multidimensional Wiener processes," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 3426–3431.

[17] W. Masri, I. Al Ridhawi, N. Mostafa, and P. Pourghomi, "Minimizing delay in IoT systems through collaborative fog-to-fog (F2F) communication," in *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2017, pp. 1005–1010.

[18] C.-H. Tsai and C.-C. Wang, "Age-of-Information revisited: Two-way delay and distribution-oblivious online algorithm," in *2020 IEEE International Symposium on Information Theory (ISIT)*. IEEE, June 2020, accepted and to appear.

[19] ——, "Unifying AoI minimization and remote estimation — Optimal sensor/controller coordination with random two-way delay," Purdue University, Tech. Rep., August 2020. [Online]. Available: https://docs.lib.purdue.edu/ecetr/758/

[20] P. Mörters and Y. Peres, *Brownian motion*. Cambridge University Press, 2010.

[21] R. P. Anderson and D. Milutinović, "A stochastic approach to Dubins vehicle tracking problems," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2801–2806, 2014.

[22] X. Lin, G. Sharma, R. R. Mazumdar, and N. B. Shroff, "Degenerate delay-capacity tradeoffs in ad-hoc networks with Brownian mobility," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2777–2784, 2006.

[23] E. G. Varthis and D. I. Fotiadis, "A comparison of stop-and-wait and go-back-N ARQ schemes for IEEE 802.11 e wireless infrared networks," *Computer communications*, vol. 29, no. 8, pp. 1015–1025, 2006.

[24] S. B. Tor and A. E. Middleditch, "Convex decomposition of simple polygons," *ACM Transactions on Graphics (TOG)*, vol. 3, no. 4, pp. 244–265, 1984.

[25] T. Z. Ornee and Y. Sun, "Sampling for remote estimation through queues: Age of information and beyond," in *2019 International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*. IEEE, 2019, pp. 1–8.

[26] R. T. Rockafellar, *Convex analysis*. Princeton university press, 1970, no. 28.