IEEE Open Journal of the Communications Society

Minimizing the Age of Information in Status Update Systems with Multiple Sources of Uncertainty

WANJA DE SOMBRE [1], FRIEDRICH PYTTEL [1], ANDREA ORTIZ [2], ANJA KLEIN [1]

[1]Communications Engineering Lab, Technical University of Darmstadt, Germany
[2]Institute of Telecommunications, Vienna University of Technology, Austria

This work has been funded by the German Research Foundation (DFG) as a part of the projects C1 and B3 within the Collaborative Research Center (CRC) 1053 - MAKI (Nr. 210487104) and has been supported by the BMBF project Open6GHub under grant 16KISKO14, by DAAD with funds from the German Federal Ministry of Education and Research (BMBF) and by the LOEWE Center emergenCity under grant LOEWE/1/12/519/03/ 05.001(0016)/72. The work of Andrea Ortiz is funded by the Vienna Science and Technology Fund (WWTF) [Grant ID: 10.47379/VRG23002].

ABSTRACT Status Update System (SUS) are monitoring applications in the Internet of Things (IoT). They are formed by a sender that monitors a remote process and sends status updates to a receiver over a wireless data channel. The goal of the sender is to find a monitoring and transmission strategy that keeps the information at the receiver fresh, i.e., that minimizes the Age of Information (AoI) at the receiver. To be able to monitor and transmit at the optimal points in time, the sender needs to accurately track the quality of the data channel and the AoI at the receiver. The quality of the data channel is a source of uncertainty, as it is unknown to the sender. In fact, there is no possibility to be absolutely certain about the quality of the data channel at any time. The AoI at the receiver is only known at the transmitter when acknowledge (ACK) or negative acknowledge (NACK) feedback signals from the receiver are successfully decoded. However, in real applications, the feedback channel is a second source of uncertainty since it is prone to errors, thus the transmission of ACK/NACK messages might fail. Additionally, the random energy harvesting process is a third source of uncertainty. This means, the monitoring and transmission decisions have to be made amidst these multiple sources of uncertainty. To overcome this challenge, we introduce the so-called belief distribution and propose a novel joint monitoring and transmission strategy at the sender based on reinforcement learning. Our new approach, termed Continual Belief Learning, exploits the belief distribution to minimize the AoI at the receiver. Through extensive numerical simulations, we show that our proposed approach yields a significantly lower average AoI compared to state-of-the-art transmission strategies for AoI minimization in SUS.

INDEX TERMS Age of Information, Belief Learning, Internet of Things, Status Update Systems

I. INTRODUCTION

MODERN Internet of Things (IoT) devices enable widespread monitoring, e.g., of remote environmental processes [1], [2] or industrial facilities [3]. Such monitoring applications of IoT devices are commonly known as Status Update Systems (SUSs). A SUS is formed by a sender and a receiver. The sender performs the monitoring and sends status updates to the receiver over a wireless communication channel. For successful monitoring, the sender must keep the status updates at the receiver fresh. The freshness of the status updates can be evaluated using different metrics depending on the considered scenario, e.g., Age of Information (AoI), Age of Incorrect Information (AoII), Peak Age of Information (PAoI), or Query Age of Information (QAoI). An extensive overview on the topic of age related metrics is given in [4].

Among the available metrics, one of the most popular is AoI. AoI was first introduced in [22], [23], and measures the time elapsed since the generation of a status update [22]. In order to keep the AoI at the receiver low, and thus information about the monitored process fresh, the sender needs to devise monitoring and transmission strategies and to track the correct reception of the transmitted status updates at the receiver. Using its monitoring and transmission strategy, the sender decides when to monitor the remote process and when to transmit status updates. The challenge in devising such strategies for SUS comes from the fact that the sender is usually battery operated. As a result, the limited available energy needs to be efficiently allocated for monitoring and

	Metric	Battery Operated Sender	Energy Harvesting	Joint Optimization of Transmission & Monitoring	Imperfect Feedback Channel	Markov Erasure Channels	Year
[5]	AoI	✓					2019
[6]	AoI	✓					2020
[7]	AoI	✓					2023
[8]	AoII	✓					2020
[9]	QAoI	✓				✓	2022
[10]	AoI	✓	✓			✓	2023
[11]	AoI	✓	✓				2023
[12]	AoI	✓	✓				2024
[13]	AoI	✓	✓				2024
[14]	AoII	✓	✓	✓			2024
[15]	AoI/AoII/QAoI	✓	✓	✓			2023
[16]	AoI	✓	✓	✓			2021
[17]	AoI	✓	✓	✓			2024
[18]	PAoI	✓	✓	✓			2021
[19]	AoI	✓	✓	✓			2021
[20]	AoI				✓		2022
[21]	AoI	✓	✓	✓	✓		2024
This work	AoI	✓	✓	✓	✓	✓	2024

TABLE 1: Summary of related work on age minimization. This work is based on our conference paper [21].

transmission in order to ensure the freshness of the status updates at the receiver. Another challenge is that the sender is uncertain about the channel state, the AoI at the receiver, and the energy harvesting process, which is necessary information to find the optimal monitoring and transmission strategy.

To illustrate the use of SUSs, consider an IoT-based environmental monitoring system, as in [2]. In this example, a battery-operated gas sensor is deployed to monitor air quality in remote or hazardous locations. The sensor detects pollutants and transmits this data over a wireless communication channel. Due to the limited energy supply and potentially unstable communication links, the system must efficiently allocate resources for monitoring and data transmission while maintaining the freshness of information.

In this work, we address scenarios like the one described above and provide a solution that optimizes the use of available energy for monitoring and transmitting status updates, despite multiple sources of uncertainty. Before we explain the concept of our solution in more detail, we first discuss the state of the art by reviewing related work on the topic.

Existing works on age based metrics in SUS focus on the design of strategies at the sender that minimize the respective age related metric at the receiver under different assumptions. Table 1 provides an overview of recent research in this area. The table contains information about the content of each individual publication, including our previous work, and compares it to this paper[1]. For each paper, we indicate by a check mark, whether it a) considers a battery operated sender, b) includes a model of an energy harvesting process at the sender, c) jointly optimizes monitoring and transmission in contrast to only optimizing transmission times, d) considers an imperfect feedback channel, and e) takes into account the existent time correlations in the data channel by modeling it as a Markov erasure channel. Additionally, we specify which age-based metric is optimized in each publication, along with the publication's respective publication date.

In [5]–[9], the AoI, QAoI, or AoII are optimized for a battery operated sender without modelling an energy harvesting process and without jointly optimizing monitoring and transmission at the sender. Instead, the authors use models with constant or random sampling processes. The authors assume that the sender operates under the constraint of a fixed and limited amount of available energy. The authors of [10]–[13] additionally model a random energy harvesting process for charging the battery. These works [5]–[13] differ in the respective optimized metric and in their

[1]Note that this journal paper is based on our conference paper [21].

data channel model, i.e., by ignoring channel correlations in time or by modeling the channel as a Markov erasure channel. In these settings, different transmission strategies are proposed depending on the amount of energy available for transmission and the knowledge the sender has about the behaviour of the SUS. Such knowledge can include the data channel quality between the sender and the receiver, or the probability of a status update generation.

Joint monitoring and transmission strategies are investigated in [14]–[19]. In this case, the sender actively decides both, when to monitor the remote process and when to transmit a status update to the receiver. In [14]–[17], the authors assume a perfect feedback channel. The authors of [18] and [19] examine two cases: one where feedback is consistently available, representing a perfect feedback channel, and another where feedback is never available, representing the complete absence of a feedback channel. They do not explore the case of an imperfect feedback channel.

The authors of [20] take a first step to investigate the impact of imperfect feedback channels in SUS. They focus on the derivation of closed-form expressions for AoI under different error models for the feedback channel. They do not consider the sender's limited energy.

In most cases, the data channel is modelled as a packet erasure channel without considering correlations in time, see [5]–[8], [11], and [14]–[21]. Only in [9] and [10], the authors consider a Markov erasure channel model, which is able to model the channel more realistically, by considering the channel's typically time-correlated behaviour.

In our previous work [21], we addressed the problem of minimizing AoI in a system with a battery-operated, energy-harvesting sender and an imperfect feedback channel modeled as a packet erasure channel. To tackle the uncertainty introduced by the feedback channel, we introduced the concept of a belief distribution, enabling the sender to estimate the AoI at the receiver based on available information.

In this work, we model the data channel as a Markov erasure channel as proposed in [24]. This model is able to capture the fact that, in real applications, the data channel quality fluctuates over time [25]. Modelling the state of the data channel using a Markov chain has two consequences: On the one hand, it allows the sender to exploit the correlation in time to find a strategy to minimize the AoI at the receiver. On the other hand, the sender is uncertain about the current state of the Markov chain modelling the channel quality and therefore, has to estimate first this state before being able to exploit this additional knowledge.

Similarly, instead of a perfect feedback channel, we model the feedback channel as a Markov erasure channel. A perfect feedback channel implies that the sender perfectly knows whether a transmitted status update is correctly received or not. Thus, it is able to accurately track the AoI at the receiver. However, in real applications the feedback channel is prone to errors and the feedback might get lost. This poses an additional challenge for the design of transmission strategies because the monitoring and transmission decisions have to be made under uncertainty about the receiver's AoI.

In addition to the uncertainty brought by the data and feedback channels, the third source of uncertainty we consider in this paper is a random energy harvesting process as in [10]–[19]. Under these uncertainty sources, we investigate the design of joint monitoring and transmission strategies that minimize the AoI in a SUS with a battery operated energy harvesting sender.

The contributions of this paper can be summarized as follows:

- To minimize the AoI at the receiver, we propose a learning-based joint monitoring and transmission strategy at the sender, termed Continual Belief Learning, which is able to handle the *multiple sources of uncertainty*. To this aim, we extend the concept of *belief distribution* introduced in [21] to additionally include the additional uncertainty the sender has about the quality of the data channel.
- We consider the time-correlated nature of the data channel to improve the sender's transmission and monitoring decisions. Specifically, we exploit the forward mechanism to track the data channel state and use this information in our Continual Belief Learning approach to adjust the sender's decisions based on the current data channel state and its prediction for future data channel states.
- Through extensive numerical simulations, we show that our proposed Continual Belief Learning approach yields a significantly lower average AoI at the receiver compared to state-of-the-art transmission strategies for AoI minimization in SUS.

Our new approach is termed *Continual Belief Learning*. *Belief Learning* as presented in [21] only learns as long as the sender is certain about the current AoI at the receiver. In our new model, this is no longer practical. Therefore, our proposed *Continual Belief Learning* is able to learn continually, even when the sender is uncertain about the current AoI at the receiver and about the current data channel state. This *continual* learning allows us to exploit the time correlations of the data channel.

The rest of the paper is organized as follows. In Sec. II we introduce the system model. The AoI minimization problem is formulated as a Markov Decision Process (MDP) in Sec. III. Our proposed solution is presented in Sec. IV, followed by numerical results in Sec. V. Sec. VI concludes the paper.

II. SYSTEM MODEL

The considered SUS is depicted in Fig. 1. It consists of a battery-operated sender, a receiver and two wireless channels connecting them, i.e., a data channel for the transmission of status updates, and a feedback channel for the transmission of acknowledge (ACK) and negative acknowledge (NACK)

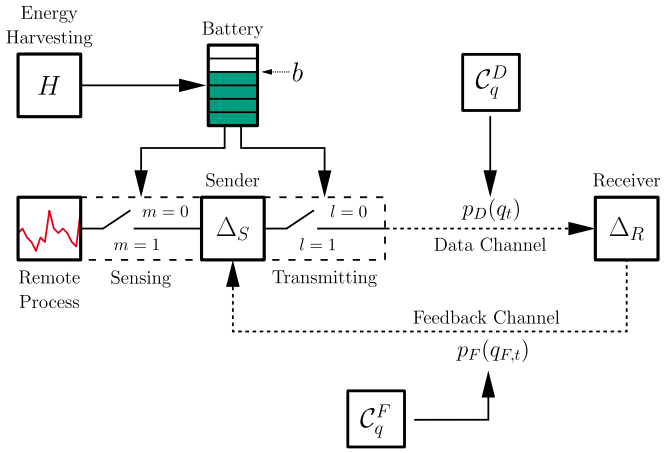

FIGURE 1: The considered SUS is formed by a battery-operated sender and a receiver.

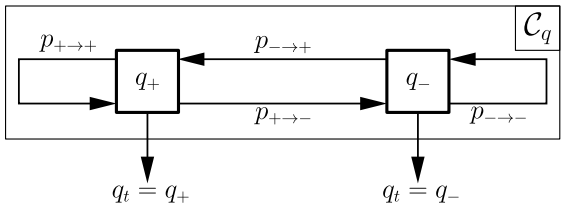

FIGURE 2: The Markov chain $\mathcal{C}_q^D$ determines the data channel quality in each time slot.

messages. We consider a time slotted system where a finite time horizon T is divided into time slots of equal length indexed by $t \in \mathbb{N}$.

In each time slot t, the sender decides on one of four possible actions. They are formed by the combination of monitoring and transmitting, denoted by $m_t, l_t \in \{0,1\}$, respectively. Every time the sender decides to monitor the remote process, i.e., $m_t = 1$, the generated status update is placed in a data buffer at the sender. The data buffer has a size of one, meaning that only the last generated status update is stored[2]. The resulting actions (m_t, l_t) are: monitor the remote process $(1,0)$, monitor the remote process and transmit the newly generated status update $(1,1)$, transmit the stored status update $(0,1)$, or remain idle $(0,0)$. The sender utilizes the energy stored in its battery to perform each of these actions. Monitoring the remote process requires $\mu \in \mathbb{N}$ energy units while transmitting a status update requires $\nu \in \mathbb{N}$ energy units. The sender's battery is assumed to have a finite capacity $B_{\max} \in \mathbb{N}$ and we denote the current battery level as $b_t \in \{0, 1, \dots, B_{\max}\}$. The battery's recharging is done through an energy harvesting process modeled by the discrete random variable H. H is uniformly distributed over the set $\mathcal{H} = \{0, 1, \dots, h_{\max}\}$ with $h_{\max} \in \mathbb{N}$. At the beginning of each time slot, a realization $h_t \in \mathcal{H}$ of H denotes the number of energy units harvested in the previous time slot. The battery level b_t denotes the total number of energy units available in time slot t and is updated in each time slot as

$$b_{t+1} = \max(0, \min(B_{\max}, b_t - m_t\mu - l_t\nu + h_t)). \quad (1)$$

The status update at the sender's data buffer is characterized by the time stamp τ_S, which indicates the time slot of its generation. In any time slot t, the AoI $\Delta_{\mathrm{S},t}$ of the status update in the sender's data buffer is defined as

$$\Delta_{\mathrm{S},t} := t - \tau_{\mathrm{S}}. \quad (2)$$

For the transmission of the status updates, i.e., $l_t = 1$, we model the wireless data channel between the sender and the receiver as a packet erasure channel. Using this model, the transmitted status update is successfully decoded at the receiver with a probability $p_{\mathrm{D},t} \in (0,1]$, which depends on the current data channel state $q_{D,t}$. We denote the probability for a successful transmission while being in state $q_{D,t}$ as $p_{\mathrm{D},t} = p_{\mathrm{D}}(q_{D,t})$.

The dynamics of the data channel state q_t are modelled as a Markov chain $\mathcal{C}_q^D$. The transition probabilities for $\mathcal{C}_q^D$ are collected in a matrix $\mathbf{A}$, where each matrix element $A_{i,j}$ is given by

$$\mathbb{P}(q_{D,t+1} = q_D^j | q_{D,t} = q_D^i) = A_{i,j} \quad \forall i,j, \quad (3)$$

where q_D^i for $i \in \{1, ..., |\mathcal{C}_q^D|\}$ are states in $\mathcal{C}_q^D$. Fig. 2 shows an example of the Markov chain when the data channel has two possible states: $q_{D,+}$, indicating a higher data channel quality and $q_{D,-}$ indicating a lower data channel quality. Typically, the probabilities to stay in the same state, i.e. $p_{+\to+}$ and $p_{-\to-}$ are considerably higher than those for changing the state, i.e. $p_{+\to-}$ and $p_{-\to+}$. This results in a bursty behaviour in the sense that errors are time-correlated. The data channel qualities for each state q_D^i are given by the matrix $\mathbf{E} = [E_{i,0}, E_{i,1}]_{i=0,\dots,|\mathcal{C}_q^D|}$, where

$$p_{\mathrm{D}}(q_D^i) = E_{i,1}, \quad (4)$$

and

$$1 - p_{\mathrm{D}}(q_D^i) = E_{i,0}. \quad (5)$$

This model can be reduced to the special case of a simple packet erasure channel by using a single state Markov chain, i.e. by setting $|\mathcal{C}_q^D| = 1$.

The receiver stores a successfully decoded status update in its data buffer. As in the sender's case, we assume the receiver's data buffer has a size of one. Similarly, we characterize the status update at the receiver's data buffer by its time stamp τ_{R}. We define the AoI $\Delta_{\mathrm{R},t}$ at the receiver as

$$\Delta_{\mathrm{R},t} := t - \tau_{\mathrm{R}}. \quad (6)$$

In every time slot, the receiver provides feedback to the sender over an imperfect feedback channel. If a status update

[2]Note that we aim at keeping the status updates fresh, so having a larger data buffer to store older status updates does not improve the performance.

is successfully decoded, the receiver transmits an ACK. If not, a NACK is sent. Mirroring the model of the data channel, we model the feedback channel as a Markov erasure channel. This means that the feedback message is successfully decoded at the sender with probability $p_{F,t} \in [0,1]$, which depends on the current state of the feedback channel $q_{F,t}$. The state $q_{F,t}$ changes according to the dynamics of a Markov chain $\mathcal{C}_q^F$.

III. PROBLEM FORMULATION

The sender's monitoring and transmission strategy allows it to decide which action (m_t, l_t) to perform in each time slot. In this section, we formulate this decision-making problem as an MDP $\mathcal{M}$. $\mathcal{M}$ is formed by a state space $\mathcal{S}$, an action space $\mathcal{A}$, a cost function c and a transition probability function P.

In time slot t, the state $s_t = (\Delta_{\mathrm{S},t}, \Delta_{\mathrm{R},t}, b_t, q_t) \in \mathcal{S}$ consists of the AoI $\Delta_{\mathrm{S},t}$ at the sender, the AoI $\Delta_{\mathrm{R},t}$ at the receiver, the sender's battery level b_t and the data channel state q_t. We consider a finite state space $\mathcal{S} := \{0,1,\dots,\Delta\} \times \{0,1,\dots,\Delta\} \times \{0,1,\dots,B_{\max}\} \times \{1,\dots,|\mathcal{C}_q^D|\}$ in which the AoI values at the sender and the receiver are limited by a maximum value Δ. We assume that old information with higher AoI than Δ has no value for the receiver. If the AoI at the sender is Δ or higher, transmission is allowed, but no longer beneficial. The state of the feedback channel is not considered as part of the system state, because although the sender's decision depends on the available feedback, it does not depend on the specific *knowledge* about the feedback channel's quality. The action space $\mathcal{A} := \{(0,0),(0,1),(1,0),(1,1)\}$ contains the sender's possible actions $a_t = (m_t, l_t)$. If the current battery level is insufficient to execute the chosen action, i.e. $b_t < \mu m_t + \nu l_t$, the sender idles instead. The cost function c assigns a cost to each state transition from s_t to s_{t+1} under an action a_t. We define the cost as

$$c(s_t, a_t, s_{t+1}) = C_t := \Delta_{\mathrm{R},t+1}. \tag{7}$$

The physical meaning of Eq. (7) is that the sender is penalized linearly with the AoI at the receiver. According to the Markov assumption, as the information at the receiver becomes older, it provides increasingly less insight into the underlying process. The transition probability function $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$ assigns a probability to every state transition under an action a_t.

A strategy $\pi \in \Pi = \mathcal{A}^{\mathcal{S}}$ is a solution of the MDP. It deterministically assigns an action a_t to every state s_t. Our goal is to design a monitoring and transmission strategy at the sender that minimizes the average AoI $\overline{\Delta}_{\mathrm{R}}$ at the receiver defined as

$$\overline{\Delta}_{\mathrm{R}} = \frac{1}{T} \sum_{i=0}^{T-1} \Delta_{\mathrm{R},i}. \tag{8}$$

The optimization problem is then given as:

$$\pi^* = \arg\min_{\pi \in \Pi} \left(\sum_{t=1}^{T} c(s_t, \pi(s_t), s_{t+1}) \right) \tag{9}$$

subject to

$$\pi(s_t) \cdot \begin{bmatrix} \mu \\ \nu \end{bmatrix} \leq b_t, \qquad \forall t = 1, \dots, T. \tag{10}$$

The optimal policy that minimizes $\overline{\Delta}_{\mathrm{R}}$ is denoted by π^*. The challenge in determining the optimal policy π^* at the sender stems from the sender's uncertainty regarding its environment. This uncertainty arises from three main factors: First, the sender's lack of knowledge about the current state of the data channel, second, the fact that the feedback is not always available since $p_{\mathrm{F}} \in (0,1]$, and third, the stochastic behaviour of the energy harvesting process. As a consequence, the sender is uncertain about the current data channel state $q_{D,t}$ and the AoI $\Delta_{\mathrm{R},t}$ at the receiver. Consequently, also $p_{D,t}$, C_t and s_t are uncertain.

IV. CONTINUAL BELIEF LEARNING

In this section, we propose a joint monitoring and transmission strategy based on reinforcement learning to find a policy that minimizes $\overline{\Delta}_{\mathrm{R}}$ under uncertainty. Our strategy, termed Continual Belief Learning, is based on the idea of building a *belief distribution* to track the state of the system in a probabilistic manner. In the following subsections, we formally define the belief distribution and describe how to update it based on monitoring and transmission decisions. Next, we present Continual Belief Learning.

A. Belief Distribution

Definition 1. *Let $\mathcal{S}$ be the state space, $\Delta \in \mathbb{N}$ be the maximum value for the AoI and $B_{\max} \in \mathbb{N}$ be the size of the battery. The belief distribution in time slot t is then defined as an array*

$$B_t \in \mathcal{B} = \mathbb{R}^{(\Delta+1)\times(\Delta+1)\times(B_{\max}+1)\times|\mathcal{C}_q^D|} \tag{11}$$

with entries $\beta^t_{i,j,k,q}$, where

$$i, j \in \{0, ..., \Delta\},$$
$$k \in \{0, ..., B_{\max}\}, \text{ and}$$
$$q \in \{1, ..., |\mathcal{C}_q^D|\}.$$

Moreover, B_t satisfies

$$\sum_{i=0}^{\Delta} \sum_{j=0}^{\Delta} \sum_{k=0}^{B_{\max}} \sum_{q=1}^{|\mathcal{C}_q^D|} \beta^t_{i,j,k,q} = 1 \tag{12}$$

and

$$\beta^t_{i,j,k,q} \in [0,1] \; \forall\, i, j, k, q. \tag{13}$$

The belief distribution B_t indicates how likely it is for the system to be in state $s_t = (\Delta_{\mathrm{S},t}, \Delta_{\mathrm{R},t}, b_t, q_{D,t})$ in time slot t given the sender's available information. The belief distribution is hence a four-dimensional tensor, where each

dimension corresponds to one variable in the state, namely Δ_{S}, Δ_{R}, b, and q_D. For a state $s = (\Delta_{\mathrm{S}}, \Delta_{\mathrm{R}}, b, q_D)$, we introduce the shorthand notation $B_t(s)$ for the entry $\beta^t_{\Delta_{\mathrm{S},t},\Delta_{\mathrm{R},t},b_t,q_{D,t}}$, which denotes the senders estimate of the probability to be in the state s_t.

In the following, we provide a detailed explanation of how B_t represents certainty and uncertainty about the state:

If the sender is completely certain about the current state in time slot t, the belief distribution B_t is concentrated in just one entry. Mathematically expressed, whenever the sender is certain about the state:

$$B_t(s_t) = 1, \tag{14}$$

and

$$B_t(s) = 0, \tag{15}$$

for each $s \neq s_t$. In general, uncertainty is expressed by a belief distribution that is *not* concentrated in a single entry.

The belief distribution can be concentrated in one or more of the dimensions without being concentrated in only one entry. In the considered system model, the sender always knows the AoI $\Delta_{\mathrm{S},t}$ at the sender and the battery state b_t. This results in B_t being concentrated in its first and third dimension. Mathematically, this translates to:

$$\beta^t_{i,j,k,q} = 0, \tag{16}$$

whenever $i \neq \Delta_{\mathrm{S},t}$ or $k \neq b_t$.

The sender is only *temporarily certain* about the AoI $\Delta_{\mathrm{R},t}$ at the receiver. $\Delta_{\mathrm{R},t}$ is known to the sender only for time slots t in which one of the following conditions is met:

- The sender transmits an update and receives an ACK feedback,
- the sender transmits an update, is certain about $\Delta_{\mathrm{R},t-1}$ and receives a NACK feedback, or
- the sender does not transmit an update but was certain about $\Delta_{\mathrm{R},t-1}$.

Every time one of these conditions is met, B_t collapses in the dimension corresponding to $\Delta_{\mathrm{R},t}$, meaning that

$$\beta^t_{i,j,k,q} = 0, \tag{17}$$

whenever $j \neq \Delta_{\mathrm{R},t}$.

If $\mathcal{C}^D_q$ consists of more than one state and if at least two of these states are visited with a probability greater than 0, the sender never receives certain information about the current data channel state $q_{D,t}$. This means, there is a *permanent uncertainty* about $q_{D,t}$ at the sender. Mathematically, this translates to B_t being distributed in its forth dimension, s.t.

$$\sum_{i=0}^{\Delta}\sum_{j=0}^{\Delta}\sum_{k=0}^{B_{\max}} \beta^t_{i,j,k,q} > 0 \tag{18}$$

for all data channel states $q \in \{1, ..., |\mathcal{C}^D_q|\}$.

Algorithm 1 Update $P^t(q_D)$

Require: Transition Matrix A,
Require: Emission Matrix E,
Require: Previous Feedback F

1: $A' \leftarrow A^\top - I$
2: $A' \leftarrow \left[A'\mathbf{1}\right]^T$
3: $b \leftarrow [0, 0, 1]^\top$
4: $\pi(q_D) \leftarrow$ Solve $A' \times \pi(q_D) = b$ ▷ estimate stat. distribution
5: $P^{t+1}(q_D) \leftarrow \pi(q_D)$ ▷ initialize probabilities
6: **for all** $f \in F$ **do**
7: **if** $f = -1$ **then** ▷ no feedback
8: $P^{t+1}(q_D) \leftarrow A \times P(q_D)$ ▷ predict next state
9: **else** ▷ With feedback f
10: $P^{t+1}(q_D) \leftarrow \mathrm{diag}(E[o]) \times P(q_D)$ ▷ update with E
11: $P^{t+1}(q_D) \leftarrow \frac{P(q_D)}{\sum_{q'_D} P(q'_D)}$ ▷ normalize over all states q'_D
12: **end if**
13: **end for**
14: **return** $P^{t+1}(q_D)$ ▷ return final state probabilities

B. Belief Distribution Update

The system's state evolves based on the arrival of system updates, the harvested energy, the data channel, the feedback channel, and the actions selected. To track the estimated probability for each state, B_t is updated in each time slot using the information available at the sender, i.e., $(m_t, l_t), b_t, \Delta_{\mathrm{S},t}$, the structure of $\mathcal{C}^D_q$, and the possibly decoded ACK/NACK feedback.

As mentioned above, the sender is uncertain about the data channel state and about the AoI at the receiver. The data channel state can first be considered separate from the uncertainty about the AoI at the receiver. This is because the data channel state has an influence on the AoI at the receiver, but not vice versa. This means, we can first find all the probabilities $P^{t+1}(q_D)$ to be in data channel state q_D in time slot $t+1$. To this end, we use a forward algorithm as described in Alg. 1.

Alg. 1 addresses the **uncertainty about the data channel state**. To this end, the algorithm first estimates the stationary distribution of the Markov chain that governs the data channel state transitions (lines 1-5). This estimate of the stationary distribution reflects the long-term behavior of the data channel, independent of the initial state.

In case even the transition matrix A and the emission matrix E of $\mathcal{C}^D_q$ are unknown, advanced estimation algorithms like the Baum-Welch algorithm [26] can be used. Based on the obtained A and E, Alg. 1 can be executed. Here, we assume that A and E are available at the sender.

Once the stationary distribution is calculated, it is used as the initial probability distribution describing the estimated probabilities for each data channel state. The algorithm then

Algorithm 2 Updating B_t

1: **if** an ACK is received **then**
2: update $\Delta_{\mathrm{R},t+1}$ ▷ Eq. (6)
3: $B_{t+1} \leftarrow (0, \dots, 0)$
4: $\beta^{t+1}_{\Delta_{\mathrm{S},t+1},\Delta_{\mathrm{R},t+1},b_{t+1},q} \leftarrow P^{t+1}(q) \quad \forall q$
▷ sender is certain about $\Delta_{\mathrm{R},t+1}$
▷ belief distribution is concentrated in a single column
5: **else**
6: $\beta^{t+1}_{i,j,k,q} \leftarrow \beta^{t}_{i,j,k,q} \; \forall i,j,k,q$
▷ old belief distribution is copied
7: **if** $m_t = 1$ **then**
8: $\beta^{t+1}_{0,j,k,q} \leftarrow \sum_{i=0}^{\Delta} \beta^{t+1}_{i,j,k,q} \quad \forall j,k,q,$
9: $\beta^{t+1}_{i,j,k,q} \leftarrow 0 \quad \forall i \in \{1,\dots,\Delta\}, j,k,q$
▷ sender monitors remote process
▷ sender is certain that $\Delta_{\mathrm{S},t+1} = 0$
10: **end if**
11: **if** $l_t = 1$ and no feedback is received **then**
12: $\beta_{i,j,k,q} \leftarrow (1-p_{\mathrm{D}}(q))\beta^{t+1}_{i,j,k,q} + \mathbb{1}_{i=j}(p_{\mathrm{D}}(q)\sum_{l=0}^{\Delta}\beta^{t+1}_{i,l,k,q})$
13: $\forall i,j,k,q$
14: $\beta^{t+1}_{i,j,k,q} \leftarrow \beta_{i,j,k,q} \; \forall i,j,k,q$
▷ sender transmits, success uncertain
▷ both outcomes are reflected in the belief distribution
15: **end if**
16: **if** $b_t \neq b_{t+1}$ **then**
17: $\beta^{t+1}_{i,j,b_{t+1},q} \leftarrow \beta^{t+1}_{i,j,b_t,q} \; \forall i,j,q,$
18: $\beta^{t+1}_{i,j,b_t,q} \leftarrow 0 \; \forall i,j,q,$
▷ battery level is updated
19: **end if**
20: **if** $\Delta_{\mathrm{S},t} \neq \Delta_{\mathrm{S},t+1}$ **then**
21: $\beta^{t+1}_{\Delta_{\mathrm{S},t+1},j,b_{t+1},q} \leftarrow \beta^{t+1}_{\Delta_{\mathrm{S},t},j,b_{t+1},q} \forall j,q$
22: $\beta^{t+1}_{\Delta_{\mathrm{S}},j,b_{t+1},q} \leftarrow 0 \; \forall j,q$
▷ AoI at the sender is increased
23: **end if**
24: $\beta^{t+1}_{i,j,k,q} \leftarrow \beta^{t+1}_{i,j-1,k,q} \quad \forall i, j \in \{0,\dots,\Delta-1\}, k,q$
25: $\beta^{t+1}_{i,\Delta,k,q} \leftarrow \beta^{t+1}_{i,\Delta,k,q} + \beta^{t}_{i,\Delta,k,q} \; \forall i,k,q$
▷ AoI at the receiver is increased
26: $\beta^{t+1}_{i,j,k,q} \leftarrow P^{t+1}(q) \cdot \beta^{t+1}_{i,j,k,q} \cdot (\sum_{i'}\sum_{j'}\sum_{k'}\beta^{t+1}_{i',j',k',q})^{-1}$
27: $\forall i,j,k,q$
▷ data channel probabilities are updated
28: **end if**
29: **return** B_{t+1} ▷ return the updated belief distribution

iteratively updates this distribution based on the feedback received in previous time steps (lines 6-13), which can be either ACK (1), NACK (0), or no reception (-1). The latter case (no reception) is used for time slots in which the sender does not attempt to transmit as well as for time slots in which no feedback is received after a transmission attempt. In both of these cases, the sender does not get any information which could be used to update the probabilities for the states of the data channel.

Alg. 1 iterates over the list F of previous feedback. For each feedback f, one of the two following cases is true:

1) **No Reception ($f = -1$):** The estimated probabilities for each data channel state are updated based on the transition probabilities of the data channel. This step reflects the *uninformed* expected evolution of the data channel state when no further information about the data channel is available.
2) **ACK/NACK ($f = 1$ or $f = 0$):** The probabilities are adjusted by the emission probabilities, which represent the likelihood of receiving a specific observation given the current data channel state. This step incorporates the received feedback to refine the estimate of the data channel's current state.

After iterating over F, the algorithm yields a probability distribution over the possible data channel states (line 14). This forward estimation approach combines the knowledge about the behavior of the data channel with real-time feedback to provide an accurate estimate of the current data channel state.

After addressing the uncertainty about the data channel state by estimating $P^{t+1}(q)$ with Alg. 1, we proceed by updating B_t. We are now able to also address the **uncertainty about the AoI at the receiver**. There are four different cases of how the uncertainty about the current AoI at the receiver evolves.

1) If an **ACK** is decoded at the sender, the sender has complete information about the AoI $\Delta_{R,t+1}$ at the receiver, irrespective of any uncertainty about $\Delta_{R,t}$ in the previous time slot.
2) If a **NACK** is decoded at the sender, the sender can deduce that the current transmission attempt was not successful and that the AoI at the receiver rises. In this case, previous uncertainty about the AoI at the receiver remains.
3) The same holds if there was **no transmission** of a status update. Previous uncertainty about the AoI at the receiver remains.
4) If there was **no reception** of a feedback, the sender has no information whether the current transmission was successful. Therefore, in addition to previous uncertainty, a new layer of uncertainty is added regarding the latest transmission attempt and the resulting AoI at the receiver.

These four cases are reflected in Alg. 2, which summarizes the update procedure of B_t.

If an ACK is decoded at the sender (line 1), $\Delta_{\mathrm{R},t+1}$ can be determined based on $\Delta_{\mathrm{S},t}$ (line 2). To update the belief distribution accordingly, we first set all entries to 0 (line 3). As the sender is certain about $\Delta_{\mathrm{S},t+1}$, $\Delta_{\mathrm{R},t+1}$, and b_{t+1}, we set the entry $\beta^{t+1}_{\Delta_{\mathrm{S},t+1},\Delta_{\mathrm{R},t+1},b_{t+1},q}$ to the previously calculated probabilities $P^{t+1}(q)$ for every q (line 4). In this case, the only remaining source of uncertainty is the data channel state.

If the sender does not receive an ACK, either because it did not attempt to transmit, or because the feedback

was lost, or because it received a NACK feedback, the belief distribution is updated using the remaining available information.

If the sender monitors, i.e., $m_t = 1$ (line 7), then $\Delta_{\mathrm{S},t+1} = 1$ because the sender has a fresh status update of the remote process. In this case, the belief distribution is updated considering that only the entries $\beta^t_{0,j,k,q}$ are non-zero (lines 8-9).

If the sender transmits a status update, i.e., $l_t = 1$, without receiving a feedback (line 11), there is a probability of $p_{\mathrm{D,t}}$ that the transmitted update was successfully received. Consequently, there is a probability of $1-p_{\mathrm{D,t}}$ that the update was not received. Both cases have to be reflected in the belief distribution. This is realized in line 12, where these two cases are visible as parts of a sum. We store the result of this sum in an intermediate variable $\beta_{i,j,k,l}$. After calculating this sum for each i, j, k and l, we set $\beta^{t+1}_{i,j,k,l}$ to $\beta_{i,j,k,l}$ (line 14).

Next, considering that the battery levels b_t and b_{t+1} and the AoI values $\Delta_{\mathrm{S},t}$ and $\Delta_{\mathrm{S},t+1}$ are perfectly known at the sender, the belief distribution is updated for these values. To this end, if the battery value changes (line 16), we copy the entry at the position of the old battery value b_t to the position of new battery value b_{t+1} (line 17). Then we set the entry at the position of the old value to 0 (line 18). The procedure for $\Delta_{S,t}$ follows in the same manner.

To consider the fact that the AoI $\Delta_{\mathrm{R},t}$ at the receiver increases in each time slot, the values of $\beta^t_{i,j,k,q}$ are shifted by one in the j^{th} dimension (line 24). In line 25, the old probabilities $\beta^t_{i,\Delta,k,q}$ are added to the new probabilities $\beta^{t+1}_{i,\Delta,k,q}$. This reflects the case that $\Delta_{R,t}$ already reached its maximum value Δ. In lines 26-27, $P^{t+1}(q)$ is used to adjust the estimated probability for the data channel state in the forth dimension of B_t. With all values of $\beta^t_{i,j,k,q}$ updated, the algorithm terminates and returns the new B_{t+1} (line 29).

C. Continual Belief Learning

In this section, we present our proposed Continual Belief Learning approach to find a strategy π^{BL} that minimizes the cumulative cost $\overline{\Delta}_{\mathrm{R}}$. In contrast to *Belief Learning* as presented in [21], this approach is able to continually learn in each time slot, even in time slots in which the sender is not certain about the current state of $\mathcal{M}$. Continual Belief Learning is based on ε-greedy Q-learning. However, and in contrast to this traditional approach, it is able to handle the uncertainty about the sender's state. To this aim, Continual Belief Learning uses a novel modified update rule for the action value function $Q(s,a)$ which exploits the belief distribution B. Our algorithm is summarized in Alg. 3.

As in standard Q-learning, Continual Belief Learning selects actions that minimize $\overline{\Delta}_{\mathrm{R}}$ based on $Q(s,a)$. The values of $Q(s,a)$ are updated according to the selected actions, the observed states, and the belief distribution B. We first initialize the learning parameters, as well as Q, and B (lines 1-2). Additionally, we initialize the state value function V (line 3). Next, we observe the initial state s_0 and update

Algorithm 3 Continual Belief Learning

1: initialize α_0, ε_0, discount factor γ
2: initialize Q and B with zeros
3: set $V(s) = \min_{a\in\mathcal{A}} Q(s,a),\ \forall s \in \mathcal{S}$ ▷ init. state value function
4: observe initial state $s = s_0$
5: initialize $P^0(q_D)$ ▷ Alg. 1
6: initialize B_0 based on s_0 ▷ Alg. 2
7: set $\pi(s) = \arg\min_{a\in\mathcal{A}} Q(s,a)\ \forall s \in \mathcal{S}$
8: **while** $t \leq T$ **do**
9: select an action $a_t = (m_t, l_t)$ ▷ ε-greedy, Eq. (19)
10: perform a_t ▷ idle, monitor and/or transmit
11: observe b_{t+1}, $\Delta_{\mathrm{S},t+1}$ and calc. $P^{t+1}(q_D)$, B_{t+1} ▷ Alg. 1, 2
12: update Q ▷ Eq. (20)
13: update $V(s) \leftarrow \min_{a\in\mathcal{A}} Q(s,a),\ \forall s \in \mathcal{S}$
14: update $\pi(s) \leftarrow \arg\min_{a\in\mathcal{A}} Q(s,a),\ \forall s \in \mathcal{S}$
15: update $B_t \leftarrow B_{t+1}$
16: **end while**

$P(q)$ and B using Alg. 1 and Alg. 2 (lines 4-6). The policy π is initialized using the state-value function Q (line 7). In every time slot t, the action $a_t = (m_t, l_t)$ is selected based on the policy π and the belief distribution B following the ε-greedy mechanism, i.e., with probability ε_t the algorithm explores by randomly selecting an action $a_t \in \mathcal{A}$, whilst with probability $1-\varepsilon_t$ the algorithm exploits the past experience by selecting the action a_t as

$$a_t = \arg\max_{a\in\mathcal{A}} \sum_{s=(\Delta_S,\Delta_R,b,q)\in\mathcal{S}} B_t(s)\, \mathbb{1}_{\pi(s)=a}. \quad (19)$$

To balance exploration and exploitation, we linearly decay ε_t over time. Using the available information at the sender, we calculate $P^{t+1}(q)$ using Alg. 1 and B_{t+1} as the update of B_t using Alg. 2 (line 11). Next, we update Q, V and π (lines 12-14). The action value function Q is updated using the belief distribution B_{t+1} as

$$\begin{aligned} Q(\hat{s}, a_t) \leftarrow & (1 - B_t(\hat{s})\alpha_t) Q(s_t, a_t) \\ & + B_t(\hat{s})\alpha_t \sum_{s'\in\mathcal{S}} B_{t+1}(s')(c(s_t, a_t, s') + \gamma V(s')) \end{aligned} \quad (20)$$

where α_t is the learning rate. By applying Eq. (20), the Q-value for each possible true state $\hat{s}$ with $B_t(\hat{s}) > 0$ is updated. The learning rate is adjusted depending on the probability that the sender is in state $\hat{s}$. Note that the next state s' is also uncertain. To account for this uncertainty, a sum over all possible s' is applied, where each s' is again weighted by its probability $B_{t+1}(s')$. The remaining parts of the equation mirror the standard variant of Q-learning.

D. Optimality

Continual Belief Learning represents a strict extension of Q-learning in the following sense: When the belief distribution remains constantly concentrated in a single entry, Continual Belief Learning behaves identically to tabular Q-learning and inherits all its properties, including guaranteed almost sure convergence under specific conditions [27], i.e., the learning rates $\alpha_t^{s,a}$ for each state s and each action a must satisfy

$$\sum_{t=1}^{\infty} \alpha_t^{s,a} = \infty, \quad \text{and} \quad \sum_{t=1}^{\infty} [\alpha_t^{s,a}]^2 < \infty. \tag{21}$$

However, when the belief distribution is not constantly concentrated, the resulting strategy is no longer guaranteed to converge to the optimal strategy as defined in Sec. III. This limitation arises because the agent, lacking certainty about the current state, must approximate the best action based on its current belief. This approximation is achieved through Continual Belief Learning.

E. Implementation and Complexity

To implement Continual Belief Learning on a simple IoT device, such as a Raspberry Pi, the range of observed values is first discretized into a set of states. For instance, in the air pollution monitoring example introduced in Sec. I, these states represent discrete ranges of pollutant concentrations. The device should be equipped to monitor the state of the process and its own battery level, and it requires a sender for transmitting status updates as well as a receiver for decoding ACK/NACK feedback. Additionally, the device needs a small memory, typically in the range of kilobytes, to store the learned transition and emission matrices A and E, past feedback F, the belief distribution B, and the Q-table.

During operation, if the transition probabilities A for data channel states and their respective emission probabilities E are not already known, the device will use the initial time steps to execute the Baum-Welch algorithm to estimate these matrices. Once A and E are determined, the device proceeds with the main loop of Alg. 3 in every time step. The required computations are lightweight and well-suited for execution on edge devices. We proceed with a detailed analysis of Continual Belief Learning's computational complexity. We analyze computational complexity in terms of time complexity, which measures the number of operations as a function of the input size, and space complexity, which quantifies the memory required relative to the input size.

The action selection in line 9 of Alg. 3 has a time complexity of $\mathcal{O}(|\mathcal{S}|) = \mathcal{O}(\Delta^2 \cdot B_{\max} \cdot |\mathcal{C}_q^D|)$, where we use Bachmann-Landau notation. The action execution in line 10 is in $\mathcal{O}(1)$. Line 11 includes Alg. 1 and Alg. 2 with time complexities of $\mathcal{O}(\max(|\mathcal{C}_q^D|, |F|))$ and $\mathcal{O}(\Delta^2 \cdot B_{\max} \cdot |\mathcal{C}_q^D|)$. Updating Q in line 12 has a time complexity of $\mathcal{O}(|\mathcal{S}|^2)$, as it requires summing over all s' for each possible true state $\hat{s}$. Lines 13 and 14 have constant time complexity, and line 15 shares the same complexity as Alg. 2. By choosing the number $|F|$ of feedbacks saved in the memory as $|F| < |\mathcal{S}|^2$, the total time complexity of Continual Belief Learning is $\mathcal{O}(|\mathcal{S}|^2) = \mathcal{O}(\Delta^4 \cdot B_{\max}^2 \cdot |\mathcal{C}_q^D|^2)$, which means that it grows quadratically in the number of states. It is important to note that this represents an upper bound, which is only reached if the belief distribution is non-zero for all possible states. In practice, this scenario rarely occurs, leading to a significantly lower computational complexity comparable to tabular Q-learning.

The space complexity of Continual Belief Learning is given as $\mathcal{O}(\max(|A|, |E|, |F|, |B|, |Q|))$. Since $|E| < |A| < |B| < |Q|$ and as it is reasonable to choose the number $|F|$ of feedbacks saved in the memory as $|F| < |Q|$, the total space complexity is $\mathcal{O}(|Q|) = \mathcal{O}(\Delta^2 \cdot B_{\max} \cdot |\mathcal{C}_q^D|)$, which is the same as for tabular Q-learning.

TABLE 2: Simulation Parameters

Parameter	Description	Value
N	number of repetitions	100
Δ	AoI cap	40
T_{learn}	no. of time steps (training)	$7.5 \cdot 10^6$
T	no. of time steps (testing)	$5 \cdot 10^4$
α_t	learning rate	$0.1 - 0.099t(T_{\text{learn}})^{-1}$
ϵ_t	prob. of random action selection	$0.9 - 0.89t(T_{\text{learn}})^{-1}$
μ	energy cost (monitoring)	3
ν	energy cost (transmission)	1
$h_{\max}$	max. harvested energy per time step	1
$B_{\max}$	battery capacity	5

V. NUMERICAL EVALUATION

A. Reference Strategies

To compare the performance of our proposed Continual Belief Learning, we consider four reference strategies.

Value Iteration: This strategy provides the optimal monitoring and transmission strategy under the assumption of a perfect feedback channel and when perfect knowledge about $\mathcal{M}$ is available. To train the Value Iteration strategy, we assume perfect knowledge about the environment $\mathcal{M}$. However after the training, *during* the simulations, the strategy only uses information which is available at the sender.

Threshold based [15]: The sender decides to jointly monitor and transmit, i.e., $(m_t, l_t) = (1, 1)$, every time the AoI at the receiver $\Delta_{\text{R},t}$ exceeds an optimal threshold as derived in [15]. In any other case, it idles.

Periodic: This strategy periodically monitors the remote process and transmits the status update $(m_t, l_t) = (1, 1)$. The period T_p is matched to the energy harvesting process, such that $T_p = \left\lceil \frac{2(\mu+\nu)}{h_{\max}} \right\rceil$.

Random: This strategy monitors the remote process and transmits the status update $(m_t, l_t) = (1, 1)$ with probability p_{R}. As in the periodic case, we match p_{R} to the energy harvesting process, such that $p_{\text{R}} = \frac{h_{\max}}{2(\mu+\nu)}$.

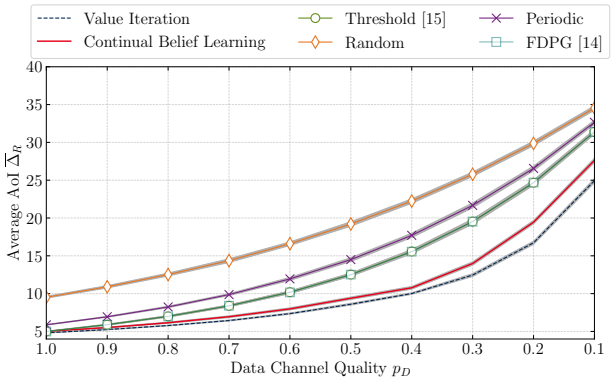

FIGURE 3: **Scenario A:** Average AoI $\overline{\Delta}_{\mathrm{R}}$ at the receiver vs. the data channel quality p_D. $p_F = 0.8$ is fixed.

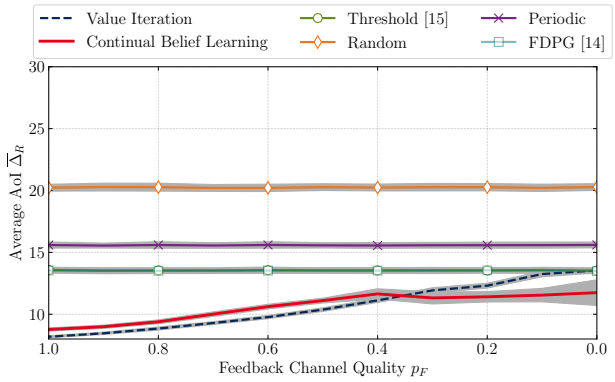

FIGURE 4: **Scenario A:** Average AoI $\overline{\Delta}_{\mathrm{R}}$ at the receiver vs. the feedback channel quality p_F. $p_D = 0.464$ is fixed.

FDPG [14]: This strategy learns the best transmission thresholds depending on the current state using a finite difference policy gradient as proposed in [14]. Every time the AoI at the receiver $\Delta_{\mathrm{R},t}$ exceeds the threshold of the current state, the sender decides to monitor and transmit, i.e., $(m_t, l_t) = (1, 1)$.

B. Simulation Setup

The considered system parameters are given in Table 2 and are used unless otherwise specified. Our proposed Continual Belief Learning is trained using $T_{\text{learn}} = 7.5 \cdot 10^6$ time slots. For the evaluation, each strategy is tested for $T = 5 \cdot 10^4$ time slots. The presented results are obtained by averaging the results of $N = 100$ independent repetitions of the simulation.

The considered Value Iteration and threshold-based approaches require a perfect feedback channel. For a fair comparison, when $p_F < 1$, we derive their respective policies π^{VI} and π^{TH} offline. These approaches build their own belief distributions B based on the information available at the sender using Alg. 1 and Alg. 2. In each time slot t, B_t is updated and the action a_t is selected based on their own policies, π^{VI} and π^{TH}, according to (19).

For the Markov chains $\mathcal{C}_q^D$ and $\mathcal{C}_q^F$, which govern the data channel quality and the feedback channel quality, we use two independent and identical Markov chains consisting of two states q_D^+ and q_D^- (q_F^+ and q_F^- respectively). Here, the states with upper index 0 indicate a channel state with higher channel quality and the states with upper index 1 indicate a state with lower data channel quality. The probability to stay in the states with upper index 0 is 0.995, which is higher than the probability to stay in the states with upper index 1, which is 0.95. This means that for 91% of the simulated time frame, both channels remain in the better state, interrupted by shorter phases with less favorable channel conditions. Accordingly, the probability to change the state are 005 and 05, respectively. We use single state Markov chains with intermediate channel qualities for initial experiments.

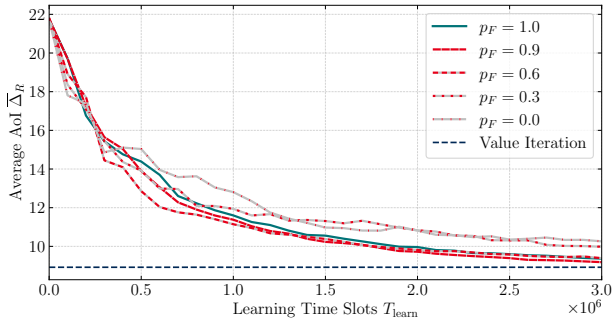

FIGURE 5: **Scenario A:** Average AoI $\overline{\Delta}_{\mathrm{R}}$ at the receiver vs. the amount of learned time slots T_{learn} of Continual Belief Learning.

We consider three different scenarios: **Scenario A**, for which both $\mathcal{C}_q^D$ and $\mathcal{C}_q^F$ each only have a single state with an intermediate channel quality, **Scenario B**, for which only the data channel varies, meaning that $\mathcal{C}_q^D$ has two states q_D^+ and q_D^-, while $\mathcal{C}_q^F$ has only one state, and **Scenario C**, for which both the data channel and the feedback channel vary, meaning that $\mathcal{C}_q^D$ and $\mathcal{C}_q^F$ both consist of two states q_D^+ and q_D^- and q_F^+ and q_F^-, respectively. For these scenarios, we first examine the results of Continual Belief Learning for different values of p_F and compare it to the reference schemes. Similarly, we compare the results for different channel qualities p_D. Additionally, we analyse the learning behaviour of Continual Belief Learning for different values of p_F and different numbers of learning time slots.

C. Simulation Results

We first investigate Scenario A, for which both Markov chains, $\mathcal{C}_q^D$ and $\mathcal{C}_q^F$ consist of a single state. This means that both, the data channel quality and the feedback channel quality are constant during the simulation. The results of Continual Belief Learning and the reference schemes are displayed in Fig. 3, 4, and 5.

Fig. 3 shows the results for Continual Belief Learning compared to all reference schemes for different data channel qualities p_D between 0.1 and 1. Here, we fixed the feedback channel quality to $p_F = 0.8$. As expected, for all strategies the average AoI at the receiver increases with decreasing data channel quality. For each data channel quality, Continual Belief Learning outperforms all non-learning reference schemes and performs close to Value Iteration. Note that Value Iteration benefits from the unrealistic assumption of having perfect knowledge of the model $\mathcal{M}$ during training. The threshold-based strategy performs best out of the three non-learning reference schemes persistently through all different data channel qualities. The FDPG strategy struggles to learn effectively due to the uncertainties in the environment. As a result, its performance remains consistently close to that of the threshold-based strategy.

The advantage of Continual Belief Learning compared to the non-learning strategies is highest for data channel qualities between $p_D = 0.5$ and $p_D = 0.2$. At $p_D = 0.4$, the advantage of Continual Belief Learning compared to the threshold-based strategy and FDPG is a 30.8% lower AoI. Compared to the periodic strategy, the advantage at $p_D = 0.4$ amounts to 39.2%. Compared to the periodic strategy, the advantage with respect to the AoI is 51.6%. At $p_D = 0.2$, we observe the highest difference of 16.3% between the performance of Continual Belief Learning and Value Iteration, which benefits from its perfect knowledge about $\mathcal{M}$. For $p_D > 0.4$, the average AoI at the receiver for Continual Belief Learning deviates at most by 10% from that for Value Iteration.

Fig. 4 shows the average AoI $\overline{\Delta}_{\mathrm{R}}$ at the receiver for different values of p_F, while the data channel quality is fixed at $p_D = 0.464$. $p_F = 1$ means that the feedback channel is perfect, $p_F = 0$ means that the sender does not receive any feedback, and $0 < p_F < 1$ means that feedback is received only intermittently. The small grey area around each of the lines represents the standard deviation of the outcomes of the $N = 100$ repetitions of the simulation.

The random strategy, the periodic strategy, the threshold-based strategy as well as FDPG are not affected by the feedback channel quality p_F. This is because the actions of the random and periodic strategy are not affected by the sender's information about the AoI at the receiver. Here, the threshold-based strategy reduces to a greedy strategy, transmitting whenever the sender has enough energy available. Therefore in these simulations, also the threshold-based strategy is not affected by the sender's knowledge about the AoI at the receiver. Hence, the respective average AoI at the receiver is constant over all examined values of p_F for all three non-learning strategies. The random strategy causes the highest average AoI at the receiver followed by the periodic strategy, the threshold-based strategy and FDPG, respectively. Value Iteration is trained for the case $p_F = 1$ and shows the best performance in this case under the unrealistic assumption that complete knowledge about the environment is available during training. As p_F decreases, the average AoI at the receiver for Value Iteration increases, indicating a continuous reduction in the performance. It reaches an average AoI comparable to that of the threshold-based approach for $p_F = 0$.

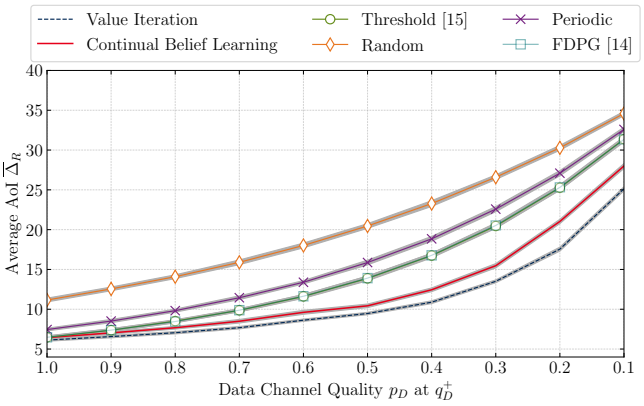

FIGURE 6: **Scenario B:** Average AoI $\overline{\Delta}_{\mathrm{R}}$ at the receiver vs. the data channel quality p_D at q_D^+.

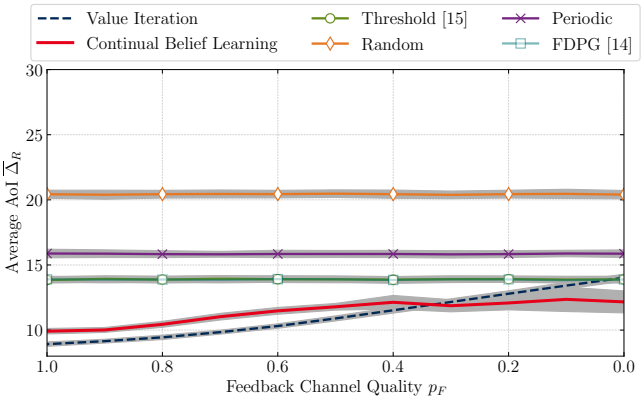

FIGURE 7: **Scenario B:** Average AoI $\overline{\Delta}_{\mathrm{R}}$ at the receiver vs. the feedback channel quality p_F^+.

For $1 \geq p_F \geq 0.4$, Continual Belief Learning performs close to Value Iteration. E.g., for $p_F = 1$, it achieves an AoI at the receiver which is only 7.2% higher than the optimal value. Moreover, for $p_F = 1$, Continual Belief Learning outperforms the threshold-based strategy by 35.4%, the periodic strategy by 43.6%, and the random strategy by 56.6%. Note that in this case, Continual Belief Learning reduces to standard Q-learning, as for a perfect feedback channel and a stationary data channel, the sender is always certain about its state. For $1 \geq p_F \geq 0.4$ the sender is able to effectively use the available feedback during learning.

For $0.4 \geq p_F \geq 0$, the available feedback is too sparse to still use it to improve the learning. Here, the sender primarily learns based on its model of the environment, which is represented by its belief distribution. Interestingly, the learning process yields excellent results when it is primarily based

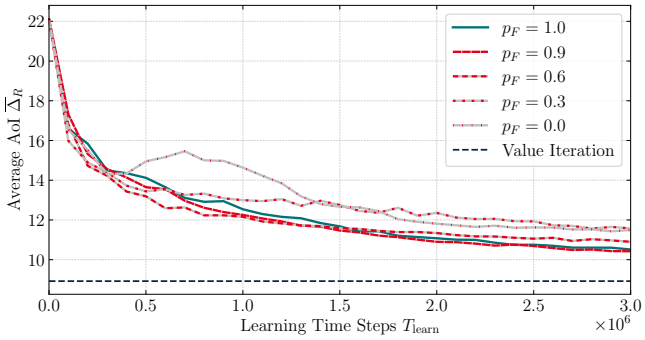

FIGURE 8: **Scenario B:** Average AoI $\overline{\Delta}_{\text{R}}$ at the receiver vs. the amount of learned time slots T_{learn} of Continual Belief Learning.

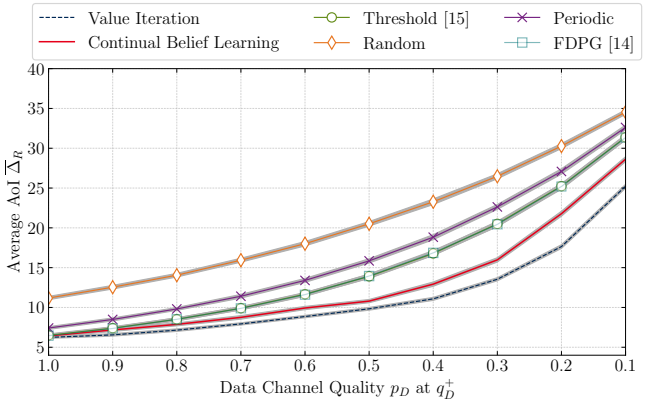

FIGURE 9: **Scenario C:** Average AoI $\overline{\Delta}_{\text{R}}$ at the receiver vs. the data channel quality p_D at q_D^+. The feedback channel qualities are $p_F(q_F^+) = 0.8$ and $p_F(q_F^-) = 0$.

on the belief distribution or even on the belief distribution alone. By using the belief distribution for learning, Continual Belief Learning is able to outperform Value Iteration without the need for complete knowledge about the environment. Continual Belief Learning outperforms Value Iteration by 13.2% at $p_F = 0$ and the threshold-based strategy, the periodic strategy and the random strategy by 13.1%, 24.7%, and 42.1%, respectively.

We evaluate the learning speed of our proposed Continual Belief Learning in Fig. 5, where we show $\overline{\Delta}_{\text{R}}$ vs. the number of learning time slots T_{learn} for different values of p_F. For every data point, we separately run Alg. 3 with that specific T_{learn}. In this way, no bias from ε-greedy occurs. Furthermore, for a fair comparison, we test the learned strategy on a system with $p_F = 1$. The performance of the Value Iteration algorithm is included at the bottom of the plot. We see that as T_{learn} increases, $\overline{\Delta}_{R,t}$ converges regardless of the value of p_F. This convergence is faster in the first 5×10^5 time slots and slows down for higher values of T_{learn}. We observe that for $p_F = 0$ and $p_F = 0.3$, the convergence of $\overline{\Delta}_{\text{R}}$ is slower compared to higher values

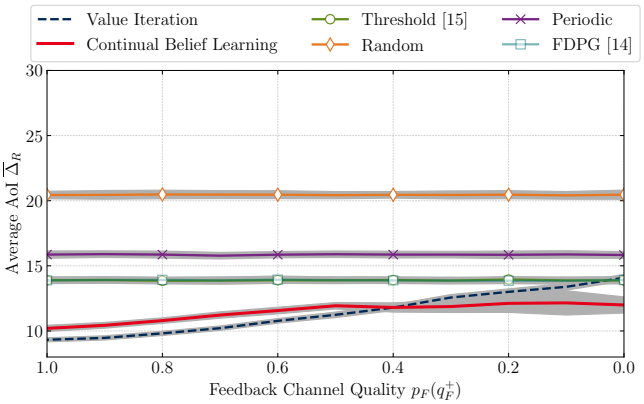

FIGURE 10: **Scenario C:** Average AoI $\overline{\Delta}_{\text{R}}$ at the receiver vs. the feedback channel quality p_F at q_F^+. The data channel qualities are $p_D(q_D^+) = 0.5$ and $p_D(q_D^-) = 0.1$.

of p_F. This is caused by higher uncertainty at the sender about the state. In contrast, for $p_F > 0.3$, the system benefits from feedback information, such that $\overline{\Delta}_{\text{R}}$ decreases faster. For $p_F = 0.6$ and for $p_F = 0.9$, the respective learning speed is higher than that for $p_F = 1$. This effect is strongest for the first learning phase and vanishes later. The reason for this observation is that by having a p_F slightly lower than 1, the sender can make use of the belief distribution. In contrast to the case $p_F = 1$, for $p_F = 0.6$ and for $p_F = 0.9$, the belief distribution does not collapse to a single entry, while the amount of available feedback is still sufficiently high. This fact allows the sender to use additional available knowledge about $\mathcal{M}$ provided by its belief distribution, which makes learning more effective.

The evaluation of Scenario B is shown in Fig. 6, 7, and 8. Now, the data channel quality is determined by the state of a binary Markov chain. However, the parameters are chosen such that over time, the average data channel quality is the same in all scenarios.

To compare the influence of different data channel qualities, we fix the data channel quality for the channel state q_D^- at 0.1 and vary the data channel quality for the channel state q_D^+, ranging from 0.1 to 1. The resulting behaviour of the average AoI at the receiver mirrors exactly the behaviour for Scenario A. Continual Belief Learning performs substantially better than all the non-learning reference schemes and FDPG. For $p_D = 0.4$, Continual Belief learning outperforms FDPG and the threshold-based strategy by 25.7% and the periodic and random strategies by 34.0% and 46.6%, respectively. The fact that the curve for Continual Belief learning in Scenario B closely resembles the curve in Scenario A means that our approach is able to perform in both cases, for a constant channel quality without time-correlated channel qualities, as well as for a bursty channel with time-correlated channel qualities.

The performance of the considered strategies for varying values of p_F is shown in Fig. 7. The observed behaviour

of Continual Belief Learning and the reference strategies is again similar to the behaviour observed in Fig. 4 for Scenario A.

The random, periodic and threshold-based strategies achieve a constant $\overline{\Delta}_{R,t}$ for all values of p_F. Again, for $p_F = 1$, Continual Belief Learning outperforms all reference strategies apart from Value Iteration. The resulting AoI at the receiver is 11.1% higher than for Value Iteration, 28.4% lower than for the threshold-based strategies, 37.6% lower than for the periodic strategy, and 51.5% lower than for the random strategy. The behaviour of Continual Belief Learning for $1 \geq p_F \geq 0.4$ is comparable to the behaviour observed for the stationary data channel case in Fig. 4. Now, even under the higher uncertainty in Scenario B, Continual Belief Learning performs better than the reference strategies for $0.4 > p_F \geq 0$. For $p_F = 0$, it achieves an AoI at the receiver that is 13.3% lower compared to Value Iteration, 12.5% lower than for the threshold-based approaches, 23.4% lower than for the periodic strategy, and 40.4% lower than for the random strategy.

The learning behaviour of Continual Belief Learning for Scenario B is shown in Fig. 8. The learning is faster for the first 5×10^4 time slots and slows down for higher values of T_{learn}. The resulting AoI at the receiver converges for every value of p_F. As in Scenario A, a feedback channel quality of $p_F = 0$ or $p_F = 0.3$ results in a slower and more volatile learning process after this initial phase. For $p_F = 0.6$ or $p_F = 0.9$, we see a small advantage in learning over the case with $p_F = 1$. Here, the same argumentation as in Scenario A applies, as for $p_F = 0.6$ and $p_F = 0.9$, Continual Belief Learning is able to exploit its additional knowledge collected in the belief distribution.

The results of simulations of Scenario C including two different states for both the data channel and the feedback channel are displayed in Fig. 9 and Fig. 10. In both figures, the general trends are comparable to the trends observed for Scenario B in Fig. 6 and Fig. 7. The reason for this is that the quality of the feedback channel, which varies in Scenario C while it was constant in Scenario B, is not part of the state of $\mathcal{M}$. This means that the sender's information about the feedback channel quality has no direct influence on the senders decision to transmit.

In Fig. 9, we show the performance of Continual Belief Learning and the reference strategies for data channel qualities between $p_D(q_D^+) = 0.1$ and $p_D(q_D^+) = 1$, while $p_D(q_D^-) = 0.1$. This time, there are two feedback channel states, q_F^+ and q_F^-. The feedback channel qualities are $p_F(q_F^+) = 0.8$ and $p_F(q_F^-) = 0$, depending on the current feedback channel state. The latter case models the possibility that in some periods, the feedback channel fails completely. The results are almost identical to the results for Scenario B. However, as expected, for all data channel qualities, Continual Belief Learning performs slightly better for Scenario B, in which the uncertainty about the channel state is lower, as the feedback channel does not experience periods with quality $p_F = 0$.

In Fig. 10, we again compare the results for different feedback channel qualities. In the state q_F^-, the feedback channel quality is 0. In the state q_F^+, the feedback channel quality is given by the values on the horizontal axis between 0 and 1. We can see the same behaviour as in Fig. 7: Continual Belief Learning performs close to Value Iteration for $1 \geq p_F(q_F^+) \geq 0.4$ and outperforms Value Iteration for smaller values of $p_F(q_F^+)$. It performs substantially better than the other reference strategies for which the threshold-based approaches have the lowest AoI at the receiver followed by the periodic approach and the random approach. At $p_F(q_F^+) = 0$, Continual Belief Learning performs 14.9% better than Value Iteration, and 13.6%, 24.2% and 41.4% better than the threshold-based, the periodic, and the random strategies, respectively.

To better understand the advantage of Continual Belief Learning and its ability of learning under uncertainty, we present the average incidences of each possible state in Scenario C in Fig. 11 for $p_F = 0$. For comparison, we display the average incidences for the Value Iteration strategy in Fig. 12. The displayed incidences are the number of occurrences of each state during the $5 \cdot 10^4$ testing time steps after training. For the states in the respective upper row, the data channel is in the better state with data channel quality $p_D = 0.5$. In the respective lower row, the data channel is in a state with lower data channel quality $p_D = 0.1$. From left to right, the battery state increases from an empty battery ($b = 0$) to a full battery ($b = 5$). In each of the small quadratic plots, on the horizontal axis we display the AoI Δ_R at the receiver. On the vertical axis, we display the AoI Δ_S at the sender. Unreachable states with a lower AoI at the receiver than at the sender are indicated by a grey filling. Reachable states which are never visited are indicated by a blue filling. States that are visited 10 or more times are indicated in red.

In both figures, we can see that states on the diagonal are frequently visited. This means that the AoI at the receiver is often the same as the AoI at the sender, matching our expectations. Furthermore, states with $p_D = 0.1$ are less frequent than states with $p_D = 0.5$, which is due to the parameters of the data channels Markov chain $\mathcal{C}_D$.

The most notable difference between the incidences for Continual Belief Learning in Fig. 11 and those for Value Iteration in Fig. 12 is that in the latter, states with higher battery levels and higher AoI at the receiver occur significantly more frequently. This suggests that Continual Belief Learning utilizes the available energy earlier than Value Iteration and avoids remaining in states with high battery levels and high AoI at the receiver. This explains its notable advantage in the case of $p_F = 0$. This is a direct result of Continual Belief Learning's ability to learn under uncertainty, a capability that Value Iteration lacks.

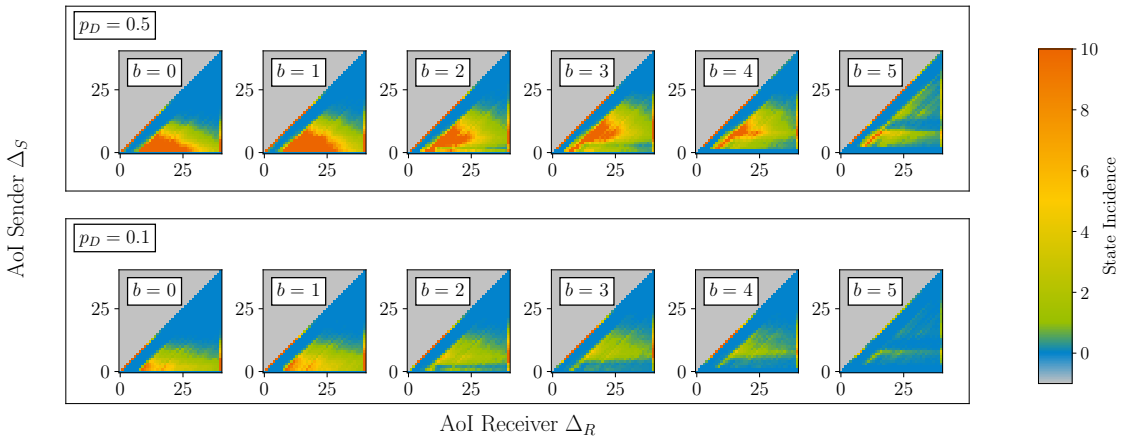

FIGURE 11: The average incidences of each state when executing the Continual Belief Learning strategy after learning in **Scenario C**.

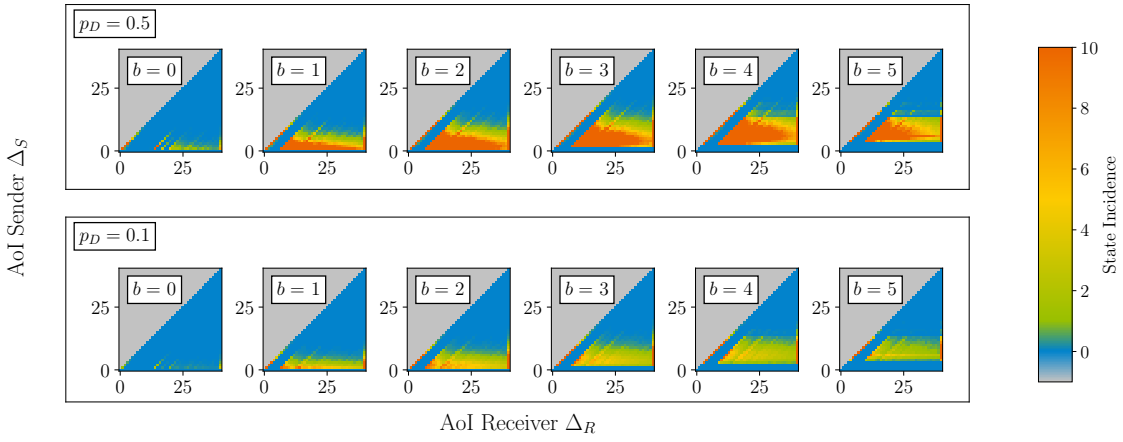

FIGURE 12: The average incidences of each state when executing the Value Iteration strategy in **Scenario C**.

VI. CONCLUSIONS

We considered a SUS in which a sender transmits status updates of a monitored process to a receiver over a wireless channel. To measure the freshness of the status update at the receiver, we considered the AoI. The optimal monitoring and transmission strategy at the sender requires knowledge about the channel state and the receiver's AoI. Knowledge about the AoI at the receiver can be obtained by means of a wireless feedback channel between receiver and sender. Considering that in real applications, the channel state is not known and the feedback channel is not perfect, we investigated the design of a monitoring and transmission strategy at the sender operating under *multiple sources of uncertainty* in the sender's environment. These sources of uncertainty are the unknown data channel state, the imperfect feedback channel and the stochastic nature of the energy harvesting process. We modeled the dynamics of the channel state using a Markov chain and estimated the current state using a forward algorithm. Furthermore, we introduced the concept of a so-called belief distribution and proposed a monitoring and transmission strategy based on reinforcement learning, termed Continual Belief Learning. We showed that Continual Belief Learning allows the sender to exploit the received ACK/NACK and the time-correlated nature of the data channel to estimate the data channel state and the receiver's AoI and make informed monitoring and transmission decisions. Through numerical simulations, we showed that Continual Belief Learning yields a lower average AoI compared to state-of-the-art transmission strategies for AoI minimization in SUS.

Future work may explore optimizing not just the AoI, but also alternative metrics like QAoI, a pull-based approach where updates are triggered by receiver requests. Additionally, integrating Continual Belief Learning with semantic communication is promising. A promising starting point is to analyse the AoII, which captures the semantic relevance of updates by quantifying the mismatch between the actual process and its perception at the receiver. On the theoretical side, an interesting direction is to further analyze the convergence properties of Continual Belief Learning, particularly to prove that it achieves optimality given the agent's available information. Finally, extending the scope from a single sender-receiver pair to network-wide systems is an interesting future direction.

REFERENCES

[1] H. Li, X. Liu, J. Li, X. Lu, and J. Huan, "Aquiculture remote monitoring system based on IoT Android platform," *Transactions of the Chinese Society of Agricultural Engineering*, 2013.

[2] S. Abraham, J. Beard, and R. Manijacob, "Remote environmental monitoring using Internet of Things (IoT)," in *2017 IEEE Global Humanitarian Technology Conference (GHTC)*, 2017.

[3] S. Adhya, D. Saha, A. Das, J. Jana, and H. Saha, "An IoT based smart solar photovoltaic remote monitoring and control unit," in *2016 2nd International Conference on Control, Instrumentation, Energy & Communication (CIEC)*, 2016.

[4] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of Information: An Introduction and Survey," *IEEE Journal on Selected Areas in Communications*, 2021.

[5] E. T. Ceran, D. Gündüz, and A. György, "Average Age of Information With Hybrid ARQ Under a Resource Constraint," *IEEE Transactions on Wireless Communications*, 2019.

[6] B. Zhou, W. Saad, M. Bennis, and P. Popovski, "Risk-Aware Optimization of Age of Information in the Internet of Things," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020.

[7] W. de Sombre, A. Ortiz, F. Aurzada, and A. Klein, "Risk-sensitive optimization and learning for minimizing age of information in point-to-point wireless communications," *IEEE International Conference on Communications (ICC)*, 2023.

[8] A. Maatouk, S. Kriouile, M. Assaad, and A. Ephremides, "The Age of Incorrect Information: A New Performance Metric for Status Updates," *IEEE/ACM Transactions on Networking*, Oct. 2020.

[9] F. Chiariotti, J. Holm, A. E. Kalør, B. Soret, S. K. Jensen, T. B. Pedersen, and P. Popovski, "Query Age of Information: Freshness in Pull-Based Communication," *IEEE Transactions on Communications*, 2022.

[10] A. Jaiswal and A. Chattopadhyay, "Age-of-information minimization for energy harvesting sensor in non-stationary environment," in *2023 21st International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, 2023.

[11] R. Zheng, M. Li, Y. Xia, Y. Ji, and X. Xu, "Aoi minimization for sensor networks with adaptive packet and energy arrival," in *2023 IEEE 24th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2023.

[12] Q. Lin, J. Su, and M. Chen, "Competitive online age-of-information optimization for energy harvesting systems," in *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications*, 2024.

[13] M. Hatami, M. Leinonen, and M. Codreanu, "Status updating under partial battery knowledge in energy harvesting iot networks," *IEEE Transactions on Green Communications and Networking*, 2024.

[14] S. Dongare, A. Jovovic, W. de Sombre, A. Ortiz, and A. Klein, "Minimizing the age of incorrect information for status update systems with energy harvesting," in *ICC 2024 - IEEE International Conference on Communications*, 2024.

[15] W. de Sombre, F. Marques, F. Pyttel, A. Ortiz, and A. Klein, "A unified approach to learn transmissionstrategies using age-based metrics inpoint-to-point wireless communication," *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, 2023.

[16] E. T. Ceran, D. Gunduz, and A. Gyorgy, "Learning to Minimize Age of Information over an Unreliable Channel with Energy Harvesting," Tech. Rep., 2021.

[17] A. Zakeri, M. Moltafet, and M. Codreanu, "Goal-oriented remote tracking of an unobservable multi-state markov source," in *2024 IEEE Wireless Communications and Networking Conference (WCNC)*, 2024.

[18] O. Ozel and P. Rafiee, "Intermittent Status Updating Through Joint Scheduling of Sensing and Retransmissions," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2021.

[19] S. Feng and J. Yang, "Age of Information Minimization for an Energy Harvesting Source With Updating Erasures: Without and With Feedback," *IEEE Transactions on Communications*, 2021.

[20] S. Rezasoltani and C. Assi, "Real-Time Status Updates in Wireless HARQ With Imperfect Feedback Channel," *IEEE Transactions on Wireless Communications*, 2022.

[21] F. Pyttel, W. de Sombre, A. Ortiz, and A. Klein, "Age of information minimization in status update systems with imperfect feedback channel," in *ICC 2024 - IEEE International Conference on Communications*, 2024.

[22] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2011.

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

[24] A. Konrad, B. Y. Zhao, A. D. Joseph, and R. Ludwig, "A markov-based channel model algorithm for wireless networks," *Wireless Networks*, 2003.

[25] G. Hasslinger and O. Hohlfeld, "The gilbert-elliott model for packet loss in real time services on the internet," in *14th GI/ITG Conference - Measurement, Modelling and Evalutation of Computer and Communication Systems*, 2008.

[26] L. E. Baum and T. Petrie, "Statistical Inference for Probabilistic Functions of Finite State Markov Chains," *The Annals of Mathematical Statistics*, 1966.

[27] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, 1992.