Minimizing the Age of Incorrect Information With Continual Belief Learning

Wanja de Sombre*, Sumedh Dongare*, Anja Klein*, Andrea Ortiz†
*Communications Engineering Lab, Technical University of Darmstadt, Germany.
†Institute of Telecommunications, Vienna University of Technology, Austria.
{w.sombre, s.dongare, a.klein}@nt.tu-darmstadt.de, andrea.ortiz@tuwien.ac.at

***Abstract*—In the context of 6G, Status Update Systems have become a pervasive component. Typically comprising a sender and a receiver, the system functions as follows: the sender observes a remote process and transmits status updates via an unreliable wireless channel. The sender's objective is to optimize the relevance and timeliness of the information received by the receiver by minimizing the Age of Incorrect Information (AoII), defined as the duration since the receiver had correct information regarding the observed process. AoII is a metric that captures both the timeliness of status updates and their semantic content. However, measuring AoII at the sender necessitates knowledge of the remote process's state at any given moment, which is only attainable if the sender constantly senses. This poses a significant challenge, particularly when sensing a new status update is energy-intensive, given the fact that the senders are small devices, often powered by energy-harvesting techniques. To address this, we propose a novel approach, Continual Belief Learning, to optimize the AoII under energy constraints. We derive a belief distribution over all possible AoII values, propose a corresponding update procedure for this distribution, and use it to learn the best sensing and transmission strategies at the sender. We validate the performance of our approach through detailed numerical simulations, using measurement data from the SKAB dataset. The simulations demonstrate the superiority of Continual Belief Learning, achieving gains of up to approximately 40% when compared to established reference schemes.**

I. Introduction

With 6G networks set to enable ubiquitous Internet of Things (IoT) connectivity, remote monitoring systems also known as Status Update Systems (SUSs) are expected to play an essential role in future digital ecosystems and cyber-physical systems. A SUS typically consists of a battery-operated sender and a receiver, and its task is to monitor a remote process. The sender is equipped with a sensor to sense the process and generate status updates. These updates are sent to the receiver over a wireless communication channel. The receiver uses the status updates to react to the changing state of the underlying monitored process. SUSs have numerous applications in 6G-enabled domains, ranging from environmental monitoring in smart cities [1] and agriculture [2], to industrial IoT [3], and healthcare systems [4].

The sender's sensing and transmission capabilities are limited by the amount of energy in its battery. Therefore, many SUSs rely on Energy Harvesting (EH) techniques to recharge the sender's battery. Using EH, the sender can exploit enviromentaly-friendy energy sources, e.g., solar, thermal, or vibrational [5]. Given the stochastic nature of these sources, the sender's battery charging follows a random process that can be effectively modeled using Markov chains.

SUSs rely on the freshness of the status updates at the receiver to enable timely responses to the changes in the monitored process. The main challenge in SUSs is to find sensing and transmission strategies for the sender to maintain this freshness, while considering the limited energy resources. A popular metric to quantify the freshness of the status updates at the receiver is the Age of Information (AoI), which measures the time elapsed since the generation of a status update [6], [7]. However, minimizing the AoI can lead to the unnecessary transmission of irrelevant information when the monitored process changes slowly, or when the monitored process returns to its previous state. To overcome these limitations, the authors of [8]–[10] introduce the Age of Incorrect Information (AoII), defined as the time elapsed since the receiver last had correct information about the monitored process. Considering the AoII allows the sender to significantly reduce its energy consumption because the status updates are transmitted only if they are relevant to the receiver. However, to calculate the AoII, the sender has to sense the monitored process in every time step such that the correctness of the information at the receiver can be evaluated. This is particularly challenging for the battery-operated senders when sensing a new status update is energy-intensive. Thus, the design of joint sensing and transmission strategies to minimize the AoII in EH SUS is a novel and unexplored research direction that is both highly relevant and timely.

Recent works have mainly focused on the design of transmission strategies to minimize the AoII when the energy required for sensing is negligible and the sender is able to sense the remote process in each time step [11]–[14]. Assuming a fixed amount of energy is available, the optimality of the threshold-based transmission strategy to minimize the AoII is proved for a two-state remote process in [11], and for remote

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]. The authors would like to thank Aleksandar Jovović and Pascal Reinhart for their valuable preliminary work.

© 2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

processes with multiple states in [12]. These approaches rely on perfect causal knowledge of the channel quality, the battery state and the AoII at the receiver, a requirement which is hard to fulfill in real applications. A threshold-based solution to minimize AoII when EH senders are considered is proposed in [13]. Although the authors overcome the strict requirement of perfect knowledge about the channel quality and the battery state, a single AoII threshold is used irrespective of the sender's available energy, which is in general suboptimal. This limitation is investigated in [14], where we propose a learning solution exploiting the threshold-based characteristic of the optimal transmission strategy for EH senders. Note, however, that sensing the remote process in each time step is still required, which may result in unnecessary sampling of the monitored process and misuse of the available energy.

On-demand sensing when the sender has a fixed amount of available energy is investigated in [15]. Specifically, the authors consider that the receiver estimates its own AoII and based on this estimation, requests the sender to sense the remote process and to transmit a status update. Even though it is a first step into reducing the number of sensing decisions, the authors aim at minimizing the estimated mean AoII and disregard the additional information about the AoII distribution. Moreover, they assume the sensing and transmission decisions are coupled. Considering them separately can lead to further reduction of the required energy.

In this paper, we propose a novel method termed *Continual Belief Learning (CBL)* to find an efficient sensing and transmission strategy that minimizes the AoII. Our approach considers sensing and transmission as separate decisions to efficiently manage the available energy. As the sender does not know the exact AoII without sensing the monitored process in every time step, the sender maintains a belief distribution over all possible AoII values and uses this distribution to learn the best sensing and transmission decisions for minimizing the AoII at the receiver. The contributions of this paper can be summarized as follows:

- We introduce the concept of a belief distribution for the AoII-minimization problem to overcome the necessity to sense in every time step. Additionally, we propose a computationally efficient update algorithm for the belief distribution. The benefits of this belief distribution are two-fold: With it, we can derive the expected current AoII, and we can accurately describe the senders knowledge about the current AoII at the receiver and use it to optimize the sender's sensing and transmission decisions.
- We employ our *CBL* algorithm to exploit the knowledge about the AoII provided by the belief distribution described above. In contrast to existing methods, e.g., Q-learning, CBL is able to learn in every time step independent of whether the sender currently knows the precise AoII or not. Moreover, by using the AoII distribution, our approach has a notable advantage over a Q-learning variant that only uses the expected AoII.
- To validate the capability of our approach, we conduct extensive numerical simulations on data taken from the SKAB test bed [16], which provides measurements of several sensors in a water circulation system. We show that our approach outperforms several reference strategies including Q-learning and performs close to the theoretical optimum derived via the value iteration algorithm.

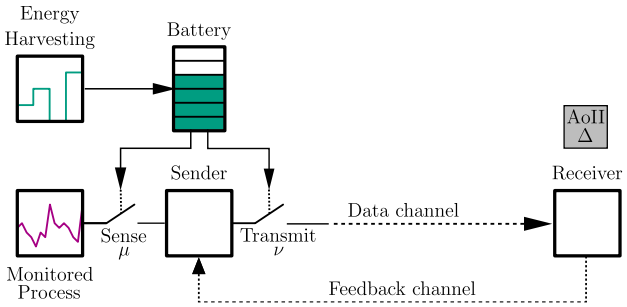

Fig. 1: System Model

The rest of this paper is organized as follows. In Sec. II the considered system model is described. In Sec. III we model the AoII minimization problem as a Markov decision process. The belief distribution update algorithm as well as our CBL approach are introduced in Sec. IV. Its performance is then analyzed in numerical simulations presented in Sec. V.

II. System Model

As shown in Fig. 1, we consider a SUS consisting of a monitored process, a sender with a battery, and a receiver. Time is divided in steps of equal duration indexed by $t \in \mathbb{N}$.

The underlying monitored process is modeled as a Markov process with N distinct states. The current state is denoted by $X_t \in \{X^1, X^2, \dots, X^N\}$ in time step t. The state transition probability $P_X(X^i, X^j) = \mathbb{P}(X_{t+1} = X^j \mid X_t = X^i)$ represents the probability of transitioning from state X^i to state X^j after time step t.

In their buffers, the sender and the receiver both store their respective most recent status update of the monitored process, which is denoted by X_t^{S} at the sender and by X_t^{R} at the receiver. X_t^{S} and X_t^{R} are also referred to as the state at the sender and the state at the receiver, respectively.

In each time step t, the sender takes one of the actions from $\mathcal{A} = \{(\mu_t, \nu_t) | \mu_t, \nu_t \in \{0, 1\}\}$, where $\mu_t = 1$ indicates that the sender senses and $\mu_t = 0$ indicates otherwise. Similarly, ν_t indicates whether the sender decides to transmit.

In case the sender idles ($\nu_t = \mu_t = 0$), $X_t^{\mathrm{S}} = X_{t-1}^{\mathrm{S}}$ and $X_t^{\mathrm{R}} = X_{t-1}^{\mathrm{R}}$. If the sender senses, $X_t^{\mathrm{S}} = X_t$. If the sender transmits, the status update is sent over a data channel which is modeled as a packet erasure channel with channel quality $p_c \in (0, 1]$. After a successful transmission, $X_t^{\mathrm{R}} = X_t^{\mathrm{S}}$. We assume that status updates are received within the same time step. If the transmission was unsuccessful, $X_t^{\mathrm{R}} = X_{t-1}^{\mathrm{R}}$. After each transmission attempt, the receiver sends either an acknowledgment (ACK) to indicate successful transmission or a negative acknowledgment (NACK) to indicate transmission failure through a feedback channel. We assume the feedback channel to be both error-free and instantaneous.

To measure the freshness of information at the receiver, we use the AoII at the receiver, which is the time elapsed since the receiver had a correct estimate of the monitored state, denoted as Δ_t and given by

$$\Delta_t = \begin{cases} 0, & \text{if } X_t = X_t^{\mathrm{R}}, \\ \min(\Delta_{t-1}+1, \Delta_{\max}), & \text{otherwise.} \end{cases} \quad (1)$$

In our model, we consider $\Delta_{\max}$ as the maximum allowable value of Δ_t before considering the status as outdated. If the status update at the receiver is outdated, the AoII at the receiver is always $\Delta_{\max}$. In order to save energy, the sender may decide not to sense in every time step and thus may not always know the exact current state of the monitored process. Consequently, the sender does not know the AoII in every time step. However, the sender knows the probability for each possible AoII-value given its knowledge about X_t^{S} and X_t^{R}. This knowledge is called the *belief distribution* at the sender.

Both, sensing and transmission consume energy from the sender's battery of finite capacity $B_{\max}$. The battery is recharged using energy harvesting. The amount of harvested energy is discrete and denoted as $E_t \in \mathcal{E}$ such that $\mathcal{E} = \{0, 1, \ldots, E_{\max}\}$. The energy harvesting process is time-correlated, i.e., the amount of harvested energy in time step t depends on the amount of harvested energy in the previous time step $t-1$. The corresponding probabilities are given by $P_E(E|E') = \mathbb{P}(E_t = E \,|\, E_{t-1} = E')$. The harvested energy in time step t is initially stored in the sender's battery, and can be used for sensing and transmitting in time step t or in later time steps. The amounts of energy required for sensing ($E^s \in \mathbb{N}$) and for transmitting ($E^{\mathrm{tx}} \in \mathbb{N}$) are assumed to be constant over time. Additionally, we assume that the amount of energy used when idling is negligible.

The battery level B_{t+1} in time step $t+1$ depends on the battery level B_t in time step t, the amount of harvested energy E_t, the battery capacity $B_{\max}$, and the action A_t taken by the sender:

$$B_{t+1} = \min\left(B_t + E_t - \mu_t \cdot E^s - \nu_t \cdot E^{\mathrm{tx}}, B_{\max}\right). \quad (2)$$

The sender is limited to select only actions for which sufficient energy is available in the battery, such that

$$B_t \geq \mu_t \cdot E^s + \nu_t \cdot E^{\mathrm{tx}}. \quad (3)$$

III. Problem Formulation

We formulate the problem as a Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, c)$. The set $\mathcal{S}$ of states contains states $S = (E, B, X, X^{\mathrm{S}}, X^{\mathrm{R}}, \Delta)$, where $E \in \mathcal{E}$ represents the harvested energy, $B \in \{0, 1, \ldots, B_{\max}\}$ denotes the battery level, $X, X^{\mathrm{S}}, X^{\mathrm{R}} \in \{X^1, X^2, \ldots, X^N\}$ the state of the monitored process, and the status updates currently stored in the sender's and the receiver's buffer, respectively. Δ denotes the AoII at the receiver. The set of actions $\mathcal{A}$ is defined as $\mathcal{A} = \{0,1\} \times \{0,1\}$, where the first component μ of the action determines whether the sender senses the monitored process and the second component ν of the action determines whether the sender transmits the currently stored status update to the receiver. The transition probability $\mathcal{P}(S, A, S') = \mathbb{P}(S_{t+1} = S' | S_t = S, A_t = A)$ is implicitly defined based on the energy harvesting process, the channel quality, and the transition probabilities of the monitored process. Note that $\mathcal{P}$ is unavailable to the sender. The cost function c is defined as $c(S, A, S') = \Delta'$, where Δ' is the AoII at the receiver in S'. The state of $\mathcal{M}$ in time step t is denoted by $S_t = (E_t, B_t, X_t, X_t^{\mathrm{S}}, X_t^{\mathrm{R}}, \Delta_t)$.

Based on this, the optimization problem is to find the optimal strategy $\pi^* : \mathcal{S} \to \mathcal{A}$, such that

$$\pi^* = \arg\min_{\pi} \lim_{T\to\infty} \frac{1}{T} \mathbb{E}\left[\sum_{t=0}^{T-1} c(S_t, A_t, S_{t+1})\right]. \quad (4)$$

IV. Proposed Solution

In this section, we describe our proposed solution to the problem formulated in Sec. III. First, we explain how we derive a belief distribution for the AoII at the sender. Second, we derive an algorithm to update this belief distribution in every time step. Third, we present *CBL*, which we use to minimize the AoII by exploiting the knowledge from the belief distribution. Please note that we assume that the sender is able to estimate the transition probabilities of the monitored process.

A. Belief Distribution

The sender does not sense the monitored process in every time step and, consequently, does not have full knowledge about the AoII at the receiver. The knowledge K_t at the sender in time step t only contains the information about the current state X_t^{S} at the sender, the current state X_t^{R} at the receiver, the current action A_t, and the knowledge $K_{t'}$ of previous time steps $t' < t$. However, the sender is able to estimate the current AoII at the receiver by using a belief distribution. This belief distribution consists of two components:

- A distribution $D_t^X : \mathcal{X} \to [0,1]$ over the states of the monitored process, indicating for every state X^i the probability to be in this state X^i in the current time step, and
- A tensor $D_t^\Delta : \mathcal{X} \times \mathcal{X} \times \{0, \ldots, \Delta\} \to [0,1]$ consisting of a probability distribution $D_t^\Delta(X, X^{\mathrm{R}})$ over all possible AoII-values for every pair of states X, X^{R} of the monitored process and at the receiver, respectively.

Given the knowledge K_t of the sender in time step t, the definitions of D_t^X and D_t^Δ are given by:

$$D_t^X(X^i) := \mathbb{P}(X_t = X^i | K_t), \text{ and} \quad (5)$$
$$D_t^\Delta(X^i, X^j, \Delta) := \mathbb{P}(\Delta_t = \Delta | X_t = X^i, X_t^{\mathrm{R}} = X^j, K_t).$$

Proposition IV.1. *Given the belief distribution (D_t^X, D_t^Δ) in a time step t, the sender is able to derive the expected AoII as follows:*

$$\mathbb{E}(\Delta_t|K_t) = \sum_{X\in\mathcal{X}} D_t^X(X) \sum_{\Delta=0}^{\Delta_{\max}} \Delta \cdot D_t^\Delta(X, X_t^R, \Delta). \quad (6)$$

Proof. The equation is a direct consequence of the definition of the belief distribution and the fact that the sender knows about the state X_t^{R} at the receiver in every time step:

$$\mathbb{E}(\Delta_t|K_t) = \sum_{\Delta=0}^{\Delta_{\max}} \Delta \cdot \mathbb{P}(\Delta_t = \Delta|K_t) \quad (7)$$

$$= \sum_{\Delta=0}^{\Delta_{\max}} \Delta \cdot \mathbb{P}(\Delta_t = \Delta|K_t, X_t^{\mathrm{R}} = X^j) \quad (8)$$

$$= \sum_{X\in\mathcal{X}} \mathbb{P}(X_t = X^i|K_t) \sum_{\Delta=0}^{\Delta_{\max}} \Delta \cdot D_t^\Delta(X, X_t^{\mathrm{R}}, \Delta) \quad (9)$$

$$= \sum_{X\in\mathcal{X}} D_t^X(X) \sum_{\Delta=0}^{\Delta_{\max}} \Delta \cdot D_t^\Delta(X, X_t^{\mathrm{R}}, \Delta) \quad (10)$$

□

Note that the distribution of all possible values of Δ_t given the knowledge K_t at the sender can be derived as

$$\mathbb{P}(\Delta_t = \Delta|K_t) = \sum_{X\in\mathcal{X}} D_t^X(X) \cdot D_t^\Delta(X, X_t^{\mathrm{R}}, \Delta). \quad (11)$$

Additionally, note that the sender can also derive further metrics from the belief distribution, such as the median AoII, the worst-case AoII, or risk-related metrics like the conditional value at risk.

In the following section, we present an algorithm to update this belief distribution in every time step.

B. Belief Distribution Update

The pseudo code for the belief distribution update is given in Alg. 1. It requires the transition probability matrix P_X of the observed process, the belief distribution $D_{t-1}^X, D_{t-1}^\Delta$ of the previous time step $t-1$ and the information μ_t whether the sender decided to sense in the current time step t.

Initially, at $t=0$, $\Delta_0 = 0$, hence $D_0^\Delta(X^i, X^j, \Delta) = \mathbf{1}_{\Delta=0}$ for $(X^i, X^j, \Delta) \in \mathcal{X}\times\mathcal{X}\times\{0,\dots,\Delta_{\max}\}$.

In lines 1-6, D_{t-1}^X is updated depending on whether the sender decides to sense, in which case the sender is certain about the current state of the monitored process, D_t collapses and becomes a unit vector (lines 2-3). In case the sender does not decide to sense, the probability for each state is updated using the transition probability matrix P_X (line 5). In the remaining lines 7-25, the tensor D_{t-1}^Δ containing the possible AoII-distributions is updated. For each state $X^i \in \mathcal{X}$, the matrix $D_{t-1}^\Delta(X^i)$ is only updated if $D_t^X(X^i) > 0$ (lines 7-8). The distribution $D_t^\Delta(X^i, X^i)$ is always known to be $[1, 0\dots, 0]$ and is not updated (line 10). In lines 11-21, the distribution $D_t^\Delta(X^i, X^j)$ is first set to $[0,\dots,0]$ and then

Algorithm 1: Belief Distribution Update

Input : Transition Matrix P_X, previous belief distributions D_{t-1}^X, D_{t-1}^Δ, decision μ_t
Output : Updated belief distributions D_t^X, D_t^Δ
1 **if** $\mu_t = 1$ **then**
2 $D_t^X \leftarrow [0,\dots,0]$
3 $D_t^X[X_{t-1}] \leftarrow 1$
4 **else**
5 $D_t^X \leftarrow D_{t-1}^X \cdot P_X$
6 **foreach** $X^i \in \mathcal{X}$ **do**
7 **if** $D_t^X(X^i) > 0$ **then**
8 **foreach** $X^j \in \mathcal{X}$ **do**
9 **if** $X^i \neq X^j$ **then**
10 $D_t^\Delta(X^i, X^j, \Delta) \leftarrow [0,\dots,0]$
11 **foreach** $X^k \in \mathcal{X}$ **do**
12 $r \leftarrow \mathrm{shift}(D_{t-1}^\Delta(X^k, X^j), 1)$
13 **if** $\nu_t = 1$ **then**
14 $D_t^\Delta(X^i, X^j) \mathrel{+}= \frac{D_{t-1}^X(X^k)}{D_t^X(X^i)} \cdot r$
15 **else**
16 $D_t^\Delta(X^i, X^j) \mathrel{+}= \frac{D_{t-1}^X(X^k)}{D_t^X(X^i)} \cdot P_X(X^k, X^i) \cdot r$
17 **return** D_t^X, D_t^Δ

incrementally updated by adding the respective share for each possible previous state $X_{t-1} = X^k$ (line 12). This share is calculated by first shifting the old distribution $D_t^\Delta(X^k, X^j)$ by one to the right to obtain r (line 13). The last entry of r is given by the sum of the last two entries of $D_t^\Delta(X^k, X^j)$. The first entry of r is 0. Depending on whether the sender decided to sense, r is then multiplied by a factor. If the sender senses in t, r is multiplied by the probability $D_{t-1}^X(X^k)$ that the observed process was in state X^k in the previous time step divided by the probability $D_t^X(X^i)$ that the monitored process is currently in state X^i. If the sender does not sense in t, r is additionally multiplied by the transition probability $P_X(X^k, X^i)$. The result is then added to the distribution $D_t^\Delta(X^i, X^j)$ (lines 14 - 17). Finally, the updated belief distribution is returned.

This proposed update algorithm maintains a constant space complexity by avoiding the need to track the full history of possible state sequences since the last sensed update from the monitored process. Instead, it aggregates all potential state sequences that could lead to each possible current state of the monitored process at every step.

C. Continual Belief Learning

Algorithm 2: Continual Belief Learning

Input : Learning rate α, discount factor γ, exploration rate ε_t, max steps $T_{\max}$
Output : Learned Q-values Q
Initialize belief distributions D_0^X, D_0^Δ
Initialize Q-table Q
for $t \leftarrow 0$ **to** $T_{\max} - 1$ **do**
 Choose action $A_t \sim \varepsilon_t$-greedy on $Q(S_t)$
 Receive new information K_{t+1} about S_{t+1} using A_t
 Update belief distributions D_{t+1}^X, D_{t+1}^Δ // See Alg. 1
 Compute estimated state distribution D_{t+1}^S for S_{t+1}
 Update Q-values Q according to Eq. 13

Using the belief distribution and Prop. IV.1, it is possible to apply the standard tabular Q-learning algorithm to solve the problem. However, standard Q-learning is not necessarily able to learn efficiently using only an estimate of the AoII. Instead, we employ *CBL*, which we developed specifically for cases in which the agent is given a probability distribution over the states of the underlying MDP. For an earlier version of this algorithm please refer to [17]. In the new version used in this work, our algorithm is capable of learning in every time step, independent of whether the current AoII is known to the sender or not.

In Alg. 2, we provide the pseudo code for CBL. It requires a learning rate α, a discount factor γ, ε_t as the probability for exploration in time step t, and the number $T_{\max}$ of learning time steps. The belief distribution is initialized as described in Sec. IV-B (line 1). Additionally, a Q-table Q is initialized with zeros (line 2). In lines 3-10, the algorithm iterates over the learning time steps $t = 0$ to $t = T_{\max} - 1$. During each time step t, the action A_t is chosen using an ε_t-greedy policy on $Q(S_t)$ (line 4). This means that with probability ε_t, a random action is chosen and with a probability $1 - \varepsilon_t$, the action is chosen according to the following equation:

$$A_t = \arg\max_{A \in \mathcal{A}} \sum_{S \in \mathcal{S}} D_t^S(S)\, \mathbf{1}_{[\arg\max_{A \in \mathcal{A}} Q(S,A) = a]}, \tag{12}$$

Next, the state of the environment is updated from S_t to S_{t+1} and the sender obtains its new knowledge K_{t+1} (line 5). It can then update its belief distribution using Alg. 1 (line 6) and computes the new estimated state distribution D_{t+1}^S based on the belief distribution (line 7). Finally, the sender updates its Q-values according to the following equation:

$$\begin{aligned} Q(\hat{S}, A_t) \leftarrow (1 - D_t^S(\hat{S})\alpha) Q(S_t, A_t) \\ + D_t^S(\hat{S})\alpha \sum_{S' \in \mathcal{S}} D_{t+1}^S(S')(-c(\hat{S}, A_t, S') + \gamma V(S')). \end{aligned} \tag{13}$$

V. Numerical Evaluation

A. Simulation Setup

We validate our approach using $N = 100$ identical repetitions of an experiment with $T_{\max} = 10\,000$ in five different status update systems referred to as scenario **A** to **E**. In scenario **A**, we train and test the strategies using a synthetically created observed process. In scenarios **B** to **E**, we train and test the strategies using measurement data from the SKAB test bed [16]. The data set provides data points from sensors installed on a water circulation system, measuring different physical quantities in every second. From these data points, we derive 10-state Markov chains for each scenario. The respective sensors we use for validation are an accelerometer (scenario **B**), a current sensor on the electric motor (**C**), a thermometer on the engine body (**D**), and a voltmeter on the motor (**E**). While scenario **C** to **E** are similar to the synthetic data, scenario **B** contains sudden fluctuations in the measurement data, making the process more difficult to predict.

In each run, we use $T_{\text{learn}} = 10\,000$ learning time steps and $T = 10\,000$ testing time steps. We further set $p_c = 0.9$, $\Delta_{\max} = 10$, $\alpha = 0.05$, $\varepsilon_t = 0.9999^t$, $E^s = E^{tx} = 1$, assuming sensing and transmitting have the same energy consumption, $B_{\max} = 10$, $\mathcal{E} = \{0, 1, 2\}$, $P_E(0,0) = 0.7$, $P_E(0, E') = P_E(E', 0) = 0.3$ for $E' \in \{1, 2\}$, and $P_E(E, E') = 0.35$ for $E, E' \in \{1, 2\}$.

We compare our approach to five reference strategies: (i) A random strategy choosing each of the four possible actions with the same probability, (ii) a greedy strategy, which always senses and transmits as soon as it has enough energy, (iii) a threshold based strategy, which senses and transmits as soon as the estimated AoII exceeds an optimal threshold (see [13]), (iv) Q-learning based on the estimated AoII, and (v) a value iteration based strategy, which serves as an upper bound. Note that value iteration eventually finds the optimal strategy, but needs the exact transition probabilities not only of the monitored process, but also of the learned MDP including the energy harvesting process and the channel quality, which are not available at the sender. Additionally, value iteration needs to know the exact value of the AoII in every time step, which is also not available at the sender.

B. Results

In Fig. 2, we display the learning behaviour of CBL and compare it to that of Q-learning in the first 1 000 learning time steps. We first average over all N runs and then plot a running mean over time. CBL is able to exploit the additional knowledge about the AoII-distribution to learn faster and to converge to a lower average AoII. The improvement in average estimated AoII as derived in Prop. IV.1 during the learning phase ranges from 9.7% for scenario **B** to 28.8% for scenario **D**. For scenario **A**, **C**, and **E**, the respective improvements are 28.3%, 15.2%, and 19.8%. Standard deviations are visualized as small shaded areas of the corresponding width in each plot. CBL significantly reduces standard deviations in all five scenarios. For scenario **A**, the average standard deviation for CBL is 39.3% lower than the standard deviation for Q-learning. For scenario **B** to **E**, the improvements in standard deviations are 48.1%, 2.4%, 38.0%, and 17.4%.

In Fig. 3, we show the average AoII of our approach compared to the five reference strategies. Standard deviations are visualized by error bars. CBL outperforms the random, greedy, threshold and Q-learning strategies and performs close to the optimal value iteration strategy, which benefits from unrealistic additional knowledge about the AoII and about the system. Note that the random strategy chooses between all four possible actions, while the threshold and greedy strategies only use the actions $(0, 0)$ and $(1, 1)$. Therefore, the performance of the random strategy surpasses the performance of the greedy and the threshold strategy in two scenarios. Furthermore, the optimal AoII threshold for transmission derived as in [13] is 0 in all five cases, reducing it to a greedy strategy, which explains that the performance of both strategies only differs because of different realizations of the random variables related to the channel and the energy harvesting process. In the synthetic scenario **A**, CBL achieves an average AoII

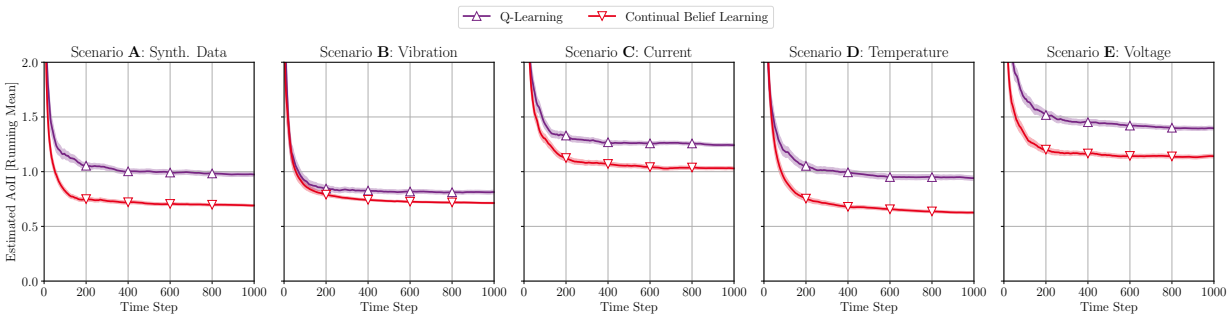

Fig. 2: Learning behaviour during the first 1 000 time steps of Q-learning and Continual Belief Learning for all scenarios.

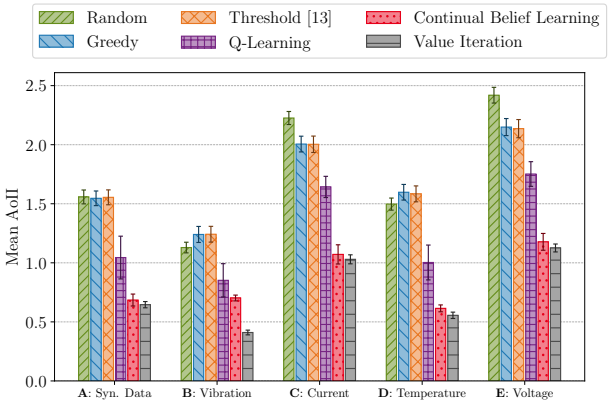

Fig. 3: Comparison of the mean AoII over 10 000 testing time steps for the reference strategies.

of 0.6850, improving the random, greedy, threshold, and Q-learning strategy by 56.0%, 55.7%, 55.9%, and 34.4%, respectively. The AoII is only 6.1% higher than the optimal AoII obtained by the value iteration strategy. For scenario **B**, improvements range from 17.6% compared to Q-learning to 43.5%, when comparing to the threshold-based strategy. The sudden fluctuations in scenario **B** make it particularly challenging to find a reasonable strategy, which can be seen in the comparably good performance of the random strategy and the comparably large gap between CBL and the optimal value. For scenario **C**, CBL achieves again 51.9%, 46.6%, 46.5%, and 34.8% lower AoII than the random, greedy, threshold, and Q-learning strategy. The difference to the optimal value is only 4.0%. The improvements in scenario **D** and **E** are again similar: In the same order, they are 59.0%, 61.5%, 61.2%, and 38.7% for scenario **D** and 51.3%, 45.2%, 44.9%, and 32.8% for scenario **E**, while the AoII for the value iterations strategy is only 10.6% and 4.6% lower.

VI. Conclusion

In this study, we examined a system comprising a sender and receiver, where the sender schedules sensing and transmission to minimize the AoII under constrained energy. As sensing at every time step incurs a cost, we introduced a belief distribution for AoII, enabling the sender to make informed decisions while conserving energy. Employing Continual Belief Learning, our approach efficiently reduces the AoII by leveraging probabilistic knowledge. Evaluations with real-world water circulation data confirmed the superior performance of our approach over reference schemes, highlighting its potential for energy-efficient 6G networks. Future work may explore further use cases of CBL in diverse environments, such as agriculture, large-scale industrial systems, or smart city sensor networks, each with distinct channel conditions. In remote areas, for example, satellite-based communication could introduce unique challenges that impact CBL's performance and require specific adaptations.

References

[1] J. Shah and B. Mishra, "Iot enabled environmental monitoring system for smart cities," in *Int. Conf. on IoT and Applications*, 2016.

[2] M. Ayaz, M. Ammad-Uddin *et al.*, "Internet-of-things-based smart agriculture: Toward making the fields talk," *IEEE Access*, 2019.

[3] J. Zhao, Y. Wang *et al.*, "Timely device status updates in industrial wireless monitoring systems under resource constraints," *IEEE IoT Journal*, 2022.

[4] R. Salama, F. Al-Turjman *et al.*, "Benefits of iot applications in health care - an overview," in *Int. Conf. on Comp. Intelligence, Commun. Tech. and Netw.*, 2023.

[5] A. Ortiz, "Optimization and learning approaches for energy harvesting wireless communication systems," 2019.

[6] S. Kaul, M. Gruteser *et al.*, "Minimizing age of information in vehicular networks," in *8th Annu. IEEE Commun. Society Conf. on Sensor, Mesh and Ad Hoc Commun. and Netw.*, 2011.

[7] R. D. Yates and S. K. Kaul, "The age of information: Real-time status updating by multiple sources," *IEEE Trans. on Inf. Theory*, 2019.

[8] A. Maatouk, S. Kriouile *et al.*, "The Age of Incorrect Information: A New Performance Metric for Status Updates," *IEEE/ACM Trans. on Netw.*, Oct. 2020.

[9] A. Maatouk, M. Assaad *et al.*, "Semantics-empowered communications through the age of incorrect information," in *IEEE Int. Conf. on Commun.*, 2022.

[10] ——, "The age of incorrect information: An enabler of semantics-empowered communication," *IEEE Trans. on Wireless Commun.*, 2023.

[11] C. Kam, S. Kompella *et al.*, "Age of incorrect information for remote estimation of a binary markov source," in *IEEE Conf. on Comp. Commun. Workshops*, 2020.

[12] Y. Chen and A. Ephremides, "Minimizing age of incorrect information for unreliable channel with power constraint," in *IEEE Global Commun. Conf.*, 2021.

[13] W. de Sombre, F. Marques *et al.*, "A unified approach to learn transmission strategies using age-based metrics in point-to-point wireless communication," in *IEEE Global Commun. Conference*, 2023.

[14] S. Dongare, A. Jovovic *et al.*, "Minimizing the age of incorrect information for status update systems with energy harvesting," in *IEEE Int. Conf. on Commun.*, 2024.

[15] S. Kriouile and M. Assaad, "Minimizing the age of incorrect information for real-time tracking of markov remote sources," in *IEEE Int. Symposium on Inf. Theory*, 2021.

[16] I. D. Katser and V. O. Kozitsin, "Skoltech anomaly benchmark," https://www.kaggle.com/dsv/1693952, 2020.

[17] F. Pyttel, W. de Sombre *et al.*, "Age of information minimization in status update systems with imperfect feedback channel," in *IEEE Int. Conf. on Commun.*, 2024.