Optimal Finite Horizon Scheduling of Wireless Networked Control Systems

Onur Ayan, *Graduate Student Member, IEEE,* Sandra Hirche, *Fellow, IEEE* Anthony Ephremides, *Life Fellow, IEEE* Wolfgang Kellerer, *Senior Member, IEEE*

***Abstract*—Control over networks is envisioned to be one of the driving applications of future mobile networks. Networked control systems contain sensors and controllers exchanging time-sensitive information to fulfill a particular control goal. In this work, we consider N heterogeneous feedback control loops closed over a wireless star network. A centralized scheduler located at the central node, i.e., base station (BS), determines the transmission schedule of sensor-to-BS and BS-to-controller communication links. We assume that each link can accommodate a single transmission at a time and is prone to data losses with time-varying probability. Moreover, each controller estimates the system state remotely based on available information. In such a setting, we formulate an optimization problem to minimize the network-induced estimation error at the controller. In particular, we determine the optimal transmission schedule on each link that leads to the minimum normalized mean squared error (nMSE) in a given finite horizon (FH). We compare the performance of our proposed FH scheduler to various schedulers from the existing literature. Our simulation results show that by solving the finite horizon problem optimally, we are able to reduce the nMSE by 10% when compared to the best performing scheduling policy among the selected policies from the state-of-the-art. Moreover, the linear-quadratic Gaussian (LQG) cost is reduced by more than 13% indicating a control performance improvement in the network.**

***Index Terms*—Networked control systems, cyber-physical systems, resource allocation, task-oriented communications, control over networks, age of information, semantics of information.**

I. Introduction

To date, wireless communication systems' evolution and standardization have been shaped around offering wider bandwidths and higher data rates to the ever-increasing demand by emerging applications and services. While the fifth generation (5G) wireless systems were marketed to be driven by the internet of everything (IoE) and smart cities, for the 6G networks, the example use cases range from extended reality services to flying vehicles, telemedicine, and connected autonomous systems [1]. However, the inevitable future bottleneck of available resources, such as spectrum and energy, drives the industry and academia to explore different solutions than adding new frequency bands, for example, concentrating on the clever and efficient utilization of already existing resources.

Wireless systems beyond 5G are envisioned to experience a fundamental paradigm shift towards an end-to-end co-design of communication, control, and computing [1]. One of the most promising and most popular ideas towards this vision is semantic and task-oriented communications (STOC) [2], [3]. The core idea of STOC lies in considering the impact and meaning of the transmitted bits on the accomplishment of a specific task. Moreover, the network's success is not measured by the amount of error-free bits communicated between the transmitter and the receiver but rather by the task-specific performance measures imposed by the communication purpose.

Networked control systems (NCSs) are feedback control loops closed over a communication network. They are one of the most prominent examples of task-oriented communications, whose communication goal is to drive the system state to the desired value through the controller's inputs when there is at least one non-ideal link in the feedback loop. Their performance is tightly coupled to the service provided by the network. A possible way to minimize the performance degradation caused by the imperfect communication is the optimal decision-making through cross-layer protocol design in the network. In fact, the convergence of control, communication, and computing is considered to be one of the driving trends of 6G wireless systems [1]. More importantly, if the network constitutes a bottleneck due to the scarcity of resources, as in many typical NCSs scenarios with multiple loops sharing a common network, it is of utmost importance that the network is able to identify and prioritize the most significant information [3] and distribute the network resources accordingly.

Radio resource management for multiple NCSs sharing a common network with limited resource availability is one of the application domains where the STOC plays a significant role. In typical scenarios, status update packets are generated frequently and regularly, whose delivery is necessary for fulfilling the underlying control tasks. However, since only a subset of these packets can be transmitted, a classification of the meaning behind the generated bits enables the network to prioritize the most urgent transmissions. This work aims to develop a mathematical framework for packet scheduling for multiple heterogeneous feedback loops where the network has a star topology. In particular, our study focuses on quantifying information in real-time closed-loop control over networks and how to use the resulting metric optimally to improve the

O. Ayan and W. Kellerer are with the Chair of Communication Networks, Technical University of Munich, Munich, Germany.
E-mail: {onur.ayan, wolfgang.kellerer}@tum.de

S. Hirche is with the Chair of Information-oriented Control, Technical University of Munich, Munich, Germany.
E-mail: hirche@tum.de

A. Ephremides is with the University of Maryland at College Park, College Park, MD, USA.
E-mail: etony@umd.edu

overall control performance. Our approach employs age of information (AoI), a metric capturing information freshness, to model the network state and to define the relationship between information freshness and the urgency to transmit. Hence, this work poses an alternative to current conventional approaches adopted by the industry that are typically oblivious to the underlying task or the context.

A. Related work

Networked control systems: NCSs have attracted many researchers from the control community for decades due to its high application potential, for example in flexible manufacturing, autonomous driving, and telemedical applications. The imperfect communication link(s) in the feedback loop makes the problem non-trivial as latency and loss of information have detrimental effect on performance and may even destabilize a system [4]. The vast majority of the existing literature focuses on the control of such systems or derives conditions for stability in the presence of a network, including but not limited to [5]–[8]. Control performance under contention-based medium access strategies has been studied in [9]–[12]. Contention-free medium access has been found to outperform random access by a numerical study in [9].

Centralized wireless resource scheduling problem for NCSs have been studied in [13]–[16]. In [16], a multi-user scenario is considered, and a new metric called *urgency of information (UoI)* is proposed. While [13], [14] propose a greedy policy, scheduling the sub-system with the highest instantaneous estimation error, [15] solves the problem for a given time horizon optimally. These works contain unrealistic network assumptions, e.g., assuming *global knowledge* for decision-making. That is, the remotely operating scheduler has global and perfect knowledge about the information content that is about to be transmitted. Moreover, [15] considers a communication link without loss or delay. Additionally, as one of the main differences to this work which assumes a star network, they consider a single-hop communication link closing the feedback loop.

Age of information and beyond: The AoI, firstly proposed by networking community in [17] and [18], emerged as a novel metric in real-time networked systems. In remote monitoring scenarios, the AoI introduced a new context and goal for the communication network: keeping the available information about the remote process fresh at the monitor. It has attracted researchers from information theory and communications societies due to its simplicity in definition and ability to disrupt conventional networking techniques maximizing throughput or minimizing delay when it comes to information freshness. AoI has been used in scenarios that include applications relying on timely and regular status updates [19]–[21]. Moreover, it has been employed as a primary metric for specific scenarios involving NCSs, such as unmanned aerial vehicles (UAVs) [22], [23] and autonomous driving [24].

Despite being a widely adopted metric in the communication community, AoI has been shown to be unsuitable for decision-making in some scenarios, as it captures only how fresh a status update is but not how valuable it is to transmit the next packet over the network. Therefore, new metrics such as the *value of information of update (VoIU)* [25] are proposed to deal with the shortcomings of AoI. To that end, *non-linear functions of AoI*, e.g., $f(\Delta(t)) = e^{a\Delta(t)}$, are employed as an enhanced metric to approximate the importance of providing the destination with new state information [25]–[27]. In [28], the authors suggest a new metric called the *age of incorrect information (AoII)* that considers an update "informative" only if the system state has changed since the latest successful transmission. The AoII follows a linear aging model when the information is incorrect.

The works mentioned above consider information semantics beyond freshness, [16], [25]–[28], e.g., VoIU, AoII, thus such functions can be seen as *approximations* of control and communications co-design. Due to their independence of system dynamics, such approaches neglect relevant aspects contributing to the true value of information for a particular task. As we are going to show later in this work the utilization of such heuristic functions lead to inadequate performance w.r.t. the underlying control tasks. On the other hand, [29]–[31] utilize metrics that are derived from the monitored process' dynamics. However, these works assume a network populated by a single user. For example, [31] focuses on control performance in a single-loop scenario and targets the sampling problem under constant loss probability and proposes a greedy policy based on the estimation error.

Preliminary results: Preliminary results of this work have been published in [32], [33], which are the most closely related works to this one. [32] proposes a greedy scheduling policy for linear time-invariant (LTI) feedback control loops under resource constraints. In [32], we do not assume any network-induced delay or loss. [33] considers packet loss modeled as a packet erasure channel with time-varying loss probability for each sub-system following a rectified Gaussian distribution. However, in [33] the scheduling problem is limited to a single-hop communication, and each received packet experiences a constant delay of one time slot. On the contrary, in this work, we tackle the uplink (UL) and downlink (DL) scheduling problem jointly while allowing arbitrary end-to-end packet delay. The introduction of the DL makes the problem statement in this work a novel and a more complex problem than the one tackled in [33]. As a solution to the complexity problem, we employ dynamic programming as a further extension to our preliminary results.

B. Main contributions

To the best of our knowledge, the joint UL and DL scheduling problem for NCSs in a multi-loop network has not been solved optimally without the assumption of global knowledge. We believe that the centralized UL and DL scheduling is relevant for future cellular networks, e.g., 6G, where sensors and controllers are connected via a base station (BS) and the limited radio resources are distributed by a scheduler located at the BS.

The key contributions of this work are as follows: *1)* We model the joint UL and DL scheduling problem of N heterogeneous feedback control loops closed over a star network as a finite horizon (FH) problem. We solve it optimally for a given horizon H. Our solution is based on the dynamic programming algorithm from [34] implemented in the form of a Monte-Carlo tree search (MCTS)[1]. In contrast to multiple prior works, our scheduling decisions do not rely on global knowledge. *2)* To achieve fairness in metric comparison, we propose normalizing MSE that simultaneously serves as a weighting factor. As a result, the scheduling metric becomes dimensionless and can be applied to a broader range of NCSs. We demonstrate its feasibility by including a realistic control system in our network. *3)* We show by simulations that adopting a control and communication co-design approach by utilizing the control-system-dependent metric, normalized MSE, leads to significant performance improvement compared to commonly used metrics in the literature, such as linear and non-linear functions of AoI. *4)* We demonstrate how one can overcome the significant computational complexity of our algorithm by applying dynamic programming for a time-memory trade-off.

The remainder of this work is outlined as follows. Section II introduces the scenario and system model considered throughout the paper. Section III briefly overviews selected heuristic solutions to the described problem. Next, our proposed solution that offers optimality in a given finite horizon H is presented in section IV. Moreover, section V shows simulation results and evaluates the effect of multiple design parameters on the final performance in detail.

C. Notation

$\mathbb{N}_0$ denotes the natural numbers including zero. The positive natural numbers and the real numbers are denoted by N^+ and $\mathbb{R}$, respectively. Throughout the paper, matrices are denoted by capital letters in bold font, i.e., $\boldsymbol{M}$, whereas small letters are used for vectors, i.e., $\boldsymbol{v}$. Additionally, $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$ denote, respectively, the n-dimensional Euclidean space and set of all $n \times m$ real matrices. Transpose of a matrix $\boldsymbol{M}$ is denoted as $\boldsymbol{M}^T$. Moreover, $\boldsymbol{M}^p$ is the p-th power of a matrix $\boldsymbol{M}$. $\Pr[A \mid B]$ denotes the conditional probability of event A given B.

II. System Model

We consider N independent control sub-systems sharing a wireless communication network[2]. Each sub-system i consists of a plant $\mathcal{P}_i$, a sensor $\mathcal{S}_i$ and a controller $\mathcal{C}_i$. The state of $\mathcal{P}_i$ is observed by $\mathcal{S}_i$ via an ideal link and transmitted over the shared wireless network to $\mathcal{C}_i$. For the simplicity of exposition, we assume that the plant state is fully observable by the sensor at all times. We assume each controller-plant pair to be co-located, whereas the sensor operates remotely.

The packets generated by each sensor $\mathcal{S}_i$ are transmitted over a wireless link to a central node called base station (BS), from where they are forwarded over a second wireless link to their respective controller $\mathcal{C}_i$. Throughout the paper, we call sensor-to-BS link *uplink (UL)* and the BS-to-controller link *downlink (DL)*. The resulting topology is referred to as a star network in the literature [36]. It resembles a topology observed in cellular networks, where each $\mathcal{S}_i$-$\mathcal{C}_i$ pair is located in the same cell of a mobile network[3]. We assume unicast communications both on the UL and the DL. This implies that each data packet contains only a single state measurement and packets of multiple users can not be concatenated along the DL. Fig. 1 depicts the considered scenario.

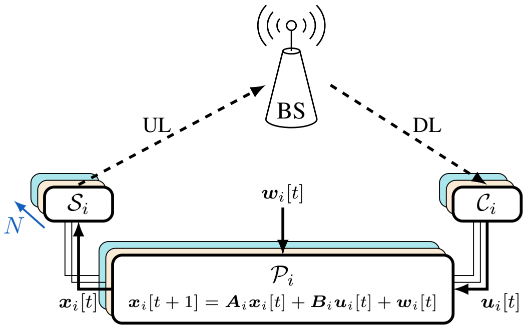
BS
UL
DL
$\mathcal{S}_i$
N
$\boldsymbol{w}_i[t]$
$\mathcal{C}_i$
$\mathcal{P}_i$
$\boldsymbol{x}_i[t+1] = \boldsymbol{A}_i\boldsymbol{x}_i[t] + \boldsymbol{B}_i\boldsymbol{u}_i[t] + \boldsymbol{w}_i[t]$
$\boldsymbol{x}_i[t]$
$\boldsymbol{u}_i[t]$

Fig. 1: N linear time invariant (LTI) control systems are closed over a shared star network. The uplink (UL) and downlink (DL) transmissions occur according to a transmission schedule determined by the central node, i.e., base station (BS). While the solid connectors represent ideal links, the wireless links prone to packet loss and delay are depicted by dashed connectors.

Time is divided into slots which is also the smallest time unit in our model. We use $t \in \mathbb{N}_0$ to index time slots and assume that each packet transmission starting in slot t ends within the same slot. Moreover, any UL packet that has been transmitted in slot t can not be sent earlier than in slot $t+1$ in the DL. It follows that any status update packet generated by $\mathcal{S}_i$ requires at least two time slots until it is successfully decoded by $\mathcal{C}_i$. Medium access is controlled by a centralized scheduler located at the BS. Each sensor $\mathcal{S}_i$ transmits only when it has been allocated a UL slot by the scheduler. In addition, the scheduler decides which packet to send in the DL if multiple packets are waiting to be forwarded. Let us introduce an indicator variable $\delta_i^{UL}[t] \in \{0,1\}$ that takes the value of 1 if $\mathcal{S}_i$ is scheduled for a UL transmission in slot t. Similarly, $\delta_i^{DL}[t] \in \{0,1\}$ indicates whether the sub-system i is scheduled for a DL transmission in slot t. The scheduler assumes that any two or more simultaneous UL transmissions, as well as DL transmissions, would fail due to collision. Therefore, if a sensor i is scheduled for a UL transmission in slot t, i.e., $\delta_i^{UL}[t] = 1$, then for any other sensor $j \neq i$, $\delta_j^{UL}[t] = 0$ holds; or equivalently $\sum_{i=1}^{N} \delta_i^{UL}[t] \leq 1$. Analogously for the DL, $\delta_i^{DL}[t] = 1$ implies $\delta_j^{DL}[t] = 0, \forall j \neq i$.

We consider time-varying UL and DL wireless qualities following the Gilbert-Elliott (GE) model [37], which is based on a two-state Markov chain, as depicted in Fig. 2. Each of the two wireless links alternate between the *good* (G), and the

[1] We refer to [35] for more information on MCTS.

[2] Here, independent means that the system dynamics of one sub-system do not affect any other sub-system..

[3] Similarly, the star network can be observed in industrial communications, where status updates are first sent to an access point and forwarded to the destination along a second link.

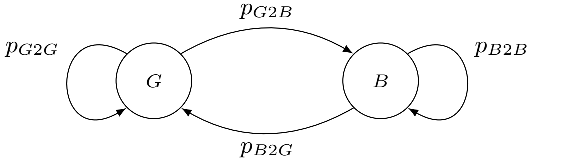

Fig. 2: Gilbert-Elliott model with good (G) and bad (B) states and their state transition probabilities. In the good state, the packet transmissions along a communication link are more likely to be successful than when in the bad state.

bad (B) states, which in return define the stationary packet loss probability in the corresponding link. That is, when in G, the failure probability of a scheduled transmission is lower than the one in B, i.e., $p_G < p_B$. If we denote the UL loss probability for sub-system i at time t as $p_i^{UL}[t]$, its behavior can be characterized by:

$$p_i^{UL}[t] = \begin{cases} p_G & \text{, if } \sigma_i^{UL}[t] = G, \\ p_B & \text{, if } \sigma_i^{UL}[t] = B, \end{cases} \tag{1}$$

where $\sigma_i^{UL}[t] \in \{G, B\}$ indicates the current state of the sensor-to-BS link. Similarly, the behavior of the BS-to-controller link can be analogously derived from (1) by substituting UL with DL everywhere. Furthermore, the transition between the G and B states occurs with a stationary probability. In particular, for any link σ_i^{XL} with $XL \in \{UL, DL\}$, the stationary state transition probabilities from G to B and from B to G are identical for all sub-systems and are defined as:

$$\begin{aligned} p_{G2B} &\triangleq \Pr\left[\sigma_i^{XL}[t] = B \mid \sigma_i^{XL}[t-1] = G\right], \forall i, \\ p_{B2G} &\triangleq \Pr\left[\sigma_i^{XL}[t] = G \mid \sigma_i^{XL}[t-1] = B\right], \forall i. \end{aligned} \tag{2}$$

Moreover, the probabilities of remaining in the same state are given as $p_{G2G} \triangleq 1 - p_{G2B}$ and $p_{B2B} \triangleq 1 - p_{B2G}$, for good and bad states, respectively. Note that σ_i^{UL} and σ_i^{DL} are two independent random variables, and the state of one link is not correlated to the state of any other link in the network. This corresponds to a scenario in which each transceiver is uniformly distributed around the cell leading to an uncorrelated channel behavior. The GE model has been widely used in the state-of-the-art to model packet loss in real-time networks, e.g., in [38], [39]. Despite its simplicity, the two-state GE model has been shown to be fairly accurate to represent Rayleigh-fading channels when the channel quality does not vary dramatically over time [40], [41][4].

$\mathcal{S}_i$ observes the system state periodically with the same speed as the slot frequency. In other words, in every slot t, a new packet is generated by the sensor[5]. Under the assumption that status is Markovian, having received an update, the controller does not benefit from receiving an older observation. Thus, older packets are considered to be obsolete and "non-informative". Therefore, $\mathcal{S}_i$ discards any older packet upon the arrival of a new update. Similarly, if the BS receives a new update from sub-system i, it discards all previously received packets from $\mathcal{S}_i$ that it has been storing to send on the DL.

A. Age of Information Model

Before introducing the control model, we formulate the AoI in the context of our system model. As our network is comprised of $2N + 1$ nodes, i.e., N sensors, N controllers, and a BS, we define the AoI from the perspective of each node separately. Let us first define the AoI for sub-system i at the BS as the time elapsed since the generation of the freshest information available at the BS. If $\nu_i^{BS}[t]$ denotes the generation time of the freshest packet the BS has received until t, then the AoI is defined as:

$$\Delta_i^{BS}[t] \triangleq t - \nu_i^{BS}[t], \tag{3}$$

where the time evolution of $\nu_i^{BS}[t]$ is characterized by:

$$\nu_i^{BS}[t+1] = \begin{cases} \nu_i^{BS}[t] & \text{, if } \quad \delta_i^{UL}[t] \cdot \gamma_i^{UL}[t] = 0, \\ t & \text{, if } \quad \delta_i^{UL}[t] \cdot \gamma_i^{UL}[t] = 1. \end{cases} \tag{4}$$

Here, $\gamma_i^{UL}[t] \in \{0, 1\}$ is the success indicator for the UL transmission in time slot t. It becomes zero in case of a failed transmission with a probability of $p_i^{UL}[t]$, given that $\mathcal{S}_i$ has been scheduled in the UL, i.e.,:

$$\Pr\left[\gamma_i^{UL}[t] = 0 \mid \delta_i^{UL}[t] = 1\right] = p_i^{UL}[t]. \tag{5}$$

Similarly, if the sub-system i is scheduled, the probability of a successful reception by the BS is given as:

$$\Pr\left[\gamma_i^{UL}[t] = 1 \mid \delta_i^{UL}[t] = 1\right] = 1 - p_i^{UL}[t]. \tag{6}$$

For the sake of completeness, we provide the probability of a successful reception by the BS under the condition that a sub-system i has not been scheduled in time slot t as zero, i.e., $\Pr\left[\gamma_i^{UL}[t] = 1 \mid \delta_i^{UL}[t] = 0\right] = 0$. It follows from the assumption that any sub-system i refrains from transmitting unless it has been granted access by the BS. Note that, in case of a successful transmission, $\nu_i^{BS}[t+1]$ drops to t. It stems from the fact that each sensor observes the system state in every slot and is able to replace any outdated information in the transmission queue with the new one. This implies that the AoI at the sensor is always zero, i.e.,:

$$\Delta_i^{\mathcal{S}_i}[t] \triangleq t - \nu_i^{\mathcal{S}_i}[t] = 0, \quad \forall i, t, \tag{7}$$

whereas the AoI at the BS is a positive natural number, i.e., $\Delta_i^{BS}[t] \in \mathbb{N}^+, \forall i, t$. In other words, the AoI at the BS followed by a successful transmission is always one, i.e.,:

$$\Delta_i^{BS}[t+1] = \begin{cases} \Delta_i^{BS}[t] + 1 & \text{, if } \quad \delta_i^{UL}[t] \cdot \gamma_i^{UL}[t] = 0, \\ 1 & \text{, if } \quad \delta_i^{UL}[t] \cdot \gamma_i^{UL}[t] = 1. \end{cases} \tag{8}$$

Suppose another indicator variable $\gamma_i^{DL}[t] \in \{0, 1\}$ that takes the value of 0 if a DL transmission in time slot t from

[4] For scenarios, in which the channel quality varies dramatically over time, finite-state Markov channel with more than two states is recommended [42].

[5] This is different than [33], in which the network is "faster" than the observation frequency, e.g., a new packet is generated once in every 3 slots. The reason not to do so in this paper is that it unnecessarily complicates the model, although it does not contribute to the core of the problem. Hence, to improve the presentation, we select equal observation and slot frequency in this work.

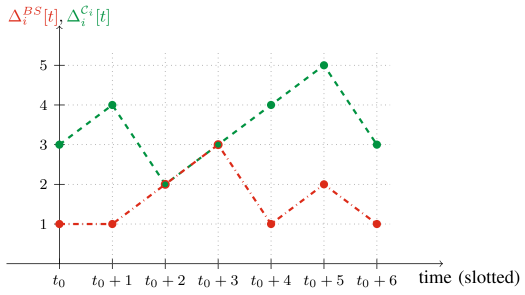

Fig. 3: A sample sequence of AoI at the BS and at the controller i. Such a sequence for $\Delta_i^{BS}[t]$ and $\Delta_i^{\mathcal{C}_i}[t]$ can be observed in our system when the UL transmissions scheduled in slots t_0-1, t_0, t_0+3, t_0+5 and the DL transmissions in slots t_0+1, t_0+5 are successful for $t \in [t_0-1, t_0+5]$, respectively.

the BS to $\mathcal{C}_i$ fails; or 1 in case it is successful. Similar to the UL case, the following equations hold for the DL:

$$\begin{aligned} &\Pr\left[\gamma_i^{DL}[t]=0 \mid \delta_i^{DL}=1\right] = p_i^{DL}[t],\\ &\Pr\left[\gamma_i^{DL}[t]=1 \mid \delta_i^{DL}=1\right] = 1-p_i^{DL}[t],\\ &\Pr\left[\gamma_i^{DL}[t]=1 \mid \delta_i^{DL}=0\right] = 0. \end{aligned} \tag{9}$$

As a result, the AoI at $\mathcal{C}_i$ can be formulated as:

$$\Delta_i^{\mathcal{C}_i}[t] = t - \nu_i^{\mathcal{C}_i}[t], \tag{10}$$

with:

$$\nu_i^{\mathcal{C}_i}[t+1] = \begin{cases} \nu_i^{\mathcal{C}_i}[t] & \text{, if} \quad \delta_i^{DL}[t]\cdot\gamma_i^{DL}[t]=0,\\ \nu_i^{BS}[t] & \text{, if} \quad \delta_i^{DL}[t]\cdot\gamma_i^{DL}[t]=1. \end{cases} \tag{11}$$

In simple words, if the controller does not receive any new information in slot t, the generation time of the freshest packet remains constant. Otherwise, upon receiving a new packet, $\nu_i^{\mathcal{C}_i}[t+1]$ is overwritten with the most recent update that is available to the BS by t. Fig. 3 illustrates a sample sequence of $\Delta_i^{BS}[t]$ and $\Delta_i^{\mathcal{C}_i}[t]$ for $t \in [t_0, t_0+6]$ following the introduced AoI model as in (3) and (10). We would like to mention that due to the star network topology, every packet available at the controller must have passed through the BS. As a result, the AoI at the BS is always less than or equal to the AoI at the controller. This implies that $\Delta_i^{BS}[t] \leq \Delta_i^{\mathcal{C}_i}[t], \forall i, t$. Moreover, as the transmission of each packet takes one time slot and we consider a two-hop network the AoI at any controller i is greater than or equal to two, i.e., $\Delta_i^{\mathcal{C}_i}[t] \geq 2, \forall i, t$.

B. Control Model

We represent the behavior of the i-th control sub-system by the following LTI model in discrete-time[6]:

$$\boldsymbol{x}_i[t+1] = \boldsymbol{A}_i\boldsymbol{x}_i[t] + \boldsymbol{B}_i\boldsymbol{u}_i[t] + \boldsymbol{w}_i[t], \tag{12}$$

with time-invariant system matrix $\boldsymbol{A}_i \in \mathbb{R}^{n_i\times n_i}$ and input matrix $\boldsymbol{B}_i \in \mathbb{R}^{n_i\times m_i}$. In addition, $\boldsymbol{w}_i[t] \in \mathbb{R}^{ni}$ represents the system noise characterized by a multi-variate Gaussian distribution with zero mean and covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{n_i\times n_i}$, i.e., $\boldsymbol{w}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_i)$. Furthermore, $\boldsymbol{x}_i[t] \in \mathbb{R}^{n_i}$ and $\boldsymbol{u}_i[t] \in \mathbb{R}^{m_i}$ denote the system state and control input, respectively. Here, the time slot t of the network corresponds to the t-th time step of each control system i. Thus, the duration of a time slot is the *sampling period* for every sub-system in our scenario. As we consider unit time in our model, the sampling period, as well as the *sampling frequency*, equals to 1 for all sub-systems. We assume that the duration of a time slot, hence the sampling frequency of each control system, is selected small enough to capture the continuous time dynamics sufficiently well.

At the beginning of each time step t, the controller $\mathcal{C}_i$ calculates $\boldsymbol{u}_i[t]$ based on the available observation history. The obtained control input is then applied to $\mathcal{P}_i$ following the dynamics from (12). Note that the input matrix $\boldsymbol{B}_i$ defines the relationship between the current control input $\boldsymbol{u}_i[t]$ and the next system state $\boldsymbol{x}_i[t+1]$. Similarly, the relationship between the current and next system states is characterized by $\boldsymbol{A}_i$.

To compensate for the shortcomings of the wireless communication network between the sensors and the controllers due to limited resources, packet loss, and delay, we consider an estimation-based controller that estimates the plant state remotely. Suppose that the freshest information available at the controller until t is given as $\boldsymbol{x}_i[t-\Delta_i^{\mathcal{C}_i}[t]]$. It can be shown that the conditional expectation of the state, which minimizes the mean squared estimation error, can be obtained from:

$$\begin{aligned} \hat{\boldsymbol{x}}_i[t] &\triangleq \mathbb{E}\left[\boldsymbol{x}_i[t] \mid \Delta_i^{\mathcal{C}_i}[t], \boldsymbol{x}_i[t-\Delta_i^{\mathcal{C}_i}[t]]\right],\\ &= \boldsymbol{A}_i^{\Delta_i^{\mathcal{C}_i}[t]}\,\boldsymbol{x}_i[t-\Delta_i^{\mathcal{C}_i}[t]] + \sum_{k=1}^{\Delta_i^{\mathcal{C}_i}[t]} \boldsymbol{A}_i^{k-1}\,\boldsymbol{B}_i\,\boldsymbol{u}_i[t-k]. \end{aligned} \tag{13}$$

The proof of (13) can be found in the supplementary document. The equation above implies that the controller has to store the last $\Delta_i^{\mathcal{C}_i}[t]$ control inputs in memory. However, it does not impose any additional communication effort as this information is already present at $\mathcal{C}_i$. We further assume that each $\mathcal{C}_i$ is aware of the time-invariant system parameters $\boldsymbol{A}_i$, $\boldsymbol{B}_i$ and $\boldsymbol{\Sigma}_i$.

The goal of the controller is to minimize the *linear-quadratic-Gaussian (LQG)* cost function $\mathcal{F}_i$:

$$\mathcal{F}_i = \limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} (\boldsymbol{x}_i[t])^T\boldsymbol{Q}_i\boldsymbol{x}_i[t] + (\boldsymbol{u}_i[t])^T\boldsymbol{R}_i\boldsymbol{u}_i[t]. \tag{14}$$

$\boldsymbol{Q}_i$ and $\boldsymbol{R}_i$ are weighting matrices of appropriate sizes that penalize the system state and control inputs in the infinite horizon. $\mathcal{F}_i$ is one of the most common metrics in control theory quantifying the control performance. The lower $\mathcal{F}_i$ is, the higher is the *quality of control (QoC)*.

We assume that the controller design is done prior to deployment; hence the characteristics of the network are unknown when the optimal control law is determined. Therefore, we obtain the optimal state feedback matrix $\boldsymbol{L}_i^* \in \mathbb{R}^{m_i\times n_i}$ as if all the links in the feedback loop were ideal. Therefore, the control law is formulated by:

$$\boldsymbol{u}_i[t] = -\boldsymbol{L}_i^*\hat{\boldsymbol{x}}_i[t], \tag{15}$$

[6] Discrete-time control systems are widely used to model digital control systems in the control theory literature [43].

corresponding to a certainty-equivalent controller that assumes a separation principle between estimation and control. The optimal state feedback matrix, minimizing the LQG cost given in (14), is obtained from:

$$\boldsymbol{L}_i^* = \left(\boldsymbol{R}_i + \boldsymbol{B}_i^T \boldsymbol{P}_i \boldsymbol{B}_i\right)^{-1} \boldsymbol{B}_i^T \boldsymbol{P}_i \boldsymbol{A}_i. \tag{16}$$

Here, $\boldsymbol{P}_i$ is a solution of the discrete time algebraic Riccati (DARE) equation given as:

$$\boldsymbol{P}_i = \boldsymbol{Q}_i + \boldsymbol{A}_i^T \big(\boldsymbol{P}_i - \boldsymbol{P}_i \boldsymbol{B}_i \big(\boldsymbol{R}_i + \boldsymbol{B}_i^T \boldsymbol{P}_i \boldsymbol{B}_i\big)^{-1} \boldsymbol{B}_i^T \boldsymbol{P}_i\big) \boldsymbol{A}_i. \tag{17}$$

Although the DARE does not take any network characteristics into account, it has been shown in [8] that the certainty-equivalence controller with the optimal controller gain $\boldsymbol{L}_i^*$ is optimal in the presence of packet loss and delay.

Let us define the network-induced estimation error as the difference between the real and estimated states:

$$\boldsymbol{e}_i[t] \triangleq \boldsymbol{x}_i[t] - \hat{\boldsymbol{x}}_i[t]. \tag{18}$$

One can easily show that by substituting (12) and (13) into (18), the instantaneous error can be expressed as a function of AoI, the system matrix and $\Delta_i^{\mathcal{C}_i}[t]$ independent noise vectors:

$$\boldsymbol{e}_i[t] = \sum_{k=1}^{\Delta_i^{\mathcal{C}_i[t]}} \boldsymbol{A}_i^{k-1} \boldsymbol{w}_i[t-k]. \tag{19}$$

In consequence, the mean squared estimation error at $\mathcal{C}_i$ can be formulated as:

$$\begin{aligned} MSE_i(\Delta_i^{\mathcal{C}_i}[t]) &\triangleq \mathbb{E}\left[(\boldsymbol{e}_i[t])^T \boldsymbol{e}_i[t]\right] \\ &= \sum_{k=1}^{\Delta_i^{\mathcal{C}_i}[t]} \operatorname{tr}\left((\boldsymbol{A}_i^T)^k \boldsymbol{A}_i^k \boldsymbol{\Sigma}_i\right) \end{aligned} \tag{20}$$

The proof can be found in the supplementary document.

We would like to mention that our proposed scheduler, which we will introduce later in section IV, uses $MSE_i(\Delta_i^{\mathcal{C}_i})$ for decision-making. This implies that as the scheduler is located at the BS, it should be able to keep track of $\Delta_i^{\mathcal{C}_i}[t]$ remotely. Nevertheless, as all packets are relayed over the BS in our considered scenario, it would be feasible for the scheduler to obtain $\Delta_i^{\mathcal{C}_i}[t]$ through a simple acknowledgment mechanism between the BS and each controller. Hence, we assume that the outcome of each DL transmissions is known to the scheduler through such an acknowledgment procedure between the controllers and the BS. Moreover, please note that the MSE also depends on $\boldsymbol{A}_i$ and $\boldsymbol{\Sigma_i}$ thus, the scheduler should be aware of these parameters for decision making. Since $\boldsymbol{A}_i$ and $\boldsymbol{\Sigma_i}$ are time-invariant matrices, a one-time information exchange prior to operation would be sufficient. It is essential to mention that the MSE does not depend on the controller gain $\boldsymbol{L}_i^*$. Therefore, the equation for MSE given AoI is independent of the controller design.

C. Problem Statement

We are interested in finding a centralized UL and DL scheduling policy $\boldsymbol{\pi}$ that aims to minimize the adverse effects induced by the (imperfect) sensor-to-controller link. In particular, the communication network's presence in the control loop creates a mismatch between the actual- and estimated state, i.e, $\boldsymbol{e}_i[t] \in \mathbb{R}^{n_i}$, rendering the controller's input obtained as (15) non-ideal. The amount of the degradation in the estimation performance is quantified by the MSE given in (20). To overcome this, we aim to keep the growth in MSE as small as possible, thereby, bring the offered QoC closer to the ideal case, i.e., when the sensor-to-controller link is a perfect communication link. To that end, we formulate the central problem of this work as:

$$\min_{\boldsymbol{\pi}} \limsup_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\frac{1}{N}\sum_{t=0}^{T-1}\sum_{i=1}^{N} c_i(\boldsymbol{e}_i[t])^T \boldsymbol{e}_i[t]\right], \tag{21}$$

with the weighting factor c_i for sub-system i. It is obvious that the optimal policy $\boldsymbol{\pi^*}$ minimizing (21) should depend on the characteristics of the communication link between each sensor-controller pair. However, in a realistic scenario, as well as in our considered network model, the wireless communication network is characterized by dynamically changing channel qualities on the UL and DL[7]. The time-varying and unpredictable nature of the network conditions does not only render solving (21) for $T \to \infty$ complex, but also sub-optimal, hence unattractive. As an alternative, we introduce a new problem that considers a look-ahead window starting from the current time slot t until $t + H$ as:

$$\min_{\boldsymbol{\pi}} \ \mathbb{E}\left[\frac{1}{N}\sum_{\tau=t}^{t+H}\sum_{i=1}^{N} g_i(\tau)\right]. \tag{22}$$

Here, $g_i(\tau)$ is a cost function in generalized form and characterizes the contribution of sub-system i to the overall cost. Note that, $c_i(\boldsymbol{e}_i[\tau])^T \boldsymbol{e}_i[\tau]$ is an example of $g_i(t)$ among other options. The selection of the parameter H controls the farsightedness of the scheduling policy $\boldsymbol{\pi}$, as it dictates the length of the *finite horizon* that the algorithm considers.

III. Greedy Uplink And Downlink Scheduling

Given the scenario and the system model introduced in section II, suppose a scheduling policy $\boldsymbol{\pi}$ that determines the next UL and DL transmissions based on the transmission history on both links. That is, at the beginning of each time slot t, the scheduler decides on $\boldsymbol{\pi}[t] = [\pi^{UL}[t] \quad \pi^{DL}[t]]^T$ with $\pi^{UL}, \pi^{DL} \in \{\varnothing, 1, 2, \ldots, N\}$ and broadcasts this information to every node in the network[8]. If a user i is scheduled in the UL, i.e., $\pi^{UL}[t] = i$, the sensor $\mathcal{S}_i$ initiates a transmission. In other words, $\pi^{UL}[t] = i$ implies $\delta_i^{UL}[t] = 1$. Similarly, $\pi^{DL}[t] = i$ means that a DL packet is sent from the BS to $\mathcal{C}_i$, i.e., $\pi^{DL}[t] = i$ and $\delta_i^{DL}[t] = 1$. If the BS does not have any new packet waiting to be forwarded, the second index of $\boldsymbol{\pi}[t]$ is an empty set, i.e., $\pi^{DL}[t] = \varnothing$.

In order to decide whether a new packet is eligible for transmission by any of the nodes, the scheduler utilizes the *age difference* between source-destination pairs in each link. For the UL case, if $\Delta_i^{BS}[t] > \Delta_i^{\mathcal{S}_i}[t] = 0$ holds for sub-system i, then $\mathcal{S}_i$ is considered to be eligible for transmission. Since

[7] This is reflected by the time-varying transmission success probabilities.

[8] We assume that the transmission of this broadcast message is instantaneous and error-free.

sensors always have a new packet to send in our considered scenario, all sensors are eligible for transmission in every time slot. However, this is not the case for the DL. For the DL, the BS may not have received new information from sub-system i since $\mathcal{C}_i$ was last updated in an earlier slot. In Fig. 3, time slots t_0+2 and t_0+3 constitute an example to such a situation, where $\Delta_i^{\mathcal{C}_i}[t] > \Delta_i^{BS}[t]$ does not hold. In that case, we refer to a DL transmission towards $\mathcal{C}_i$ as ineligible; or equivalently, $\Delta_i^{\mathcal{C}_i}[t] = \Delta_i^{BS}[t]$ implies $\pi^{DL}[t] \neq i$. If none of the sub-systems is eligible in slot t, then the BS does not send any packet in the DL, i.e., $\pi^{DL}[t] = \varnothing$.

The scheduling decisions are tightly coupled with the estimation and control performances in the network. As described in sections II-A and II-B, depending on the scheduling policy, the AoI follows a different trajectory which in return affects the network-induced estimation error. Furthermore, the estimation accuracy defines the degree of sub-optimality of control inputs that are obtained by using the state estimate instead of the actual state, as in (15)[9]. Therefore, depending on how the wireless network resources are distributed among multiple users, defined by the selected scheduling policy $\boldsymbol{\pi}$, the network can directly affect control performance metrics, such as LQG cost and MSE from (14) and (20), respectively.

As an alternative to contention-free protocols, contention-based protocols from the existing literature may also come into consideration, e.g., ALOHA [44], slotted ALOHA [45], and age-dependent random access protocol [46]. However, despite usually requiring less implementation effort due to their decentralized nature, they are known for low throughput in multi-user scenarios due to their high packet loss rate. To name an example, in a different work involving real-life measurements, we have shown that simple contention-free protocols outperform the random access protocols mentioned above w.r.t. information freshness and control performance [47]. Therefore, we do not consider contention-based protocols in the remainder of this work.

A. Application-Unaware Greedy Policies

Let us first consider two greedy scheduling policies that are widely studied in wireless communications research: *1) round-robin (RR)* and *2) maximum throughput (MT)*. They belong to the class of application-unaware policies as their operation does not depend on application layer metrics, such as AoI and MSE.

1) Round-Robin (RR) Policy: RR is one of the most frequently deployed scheduling policies in industry due to its ease of implementation. It has been used in cellular networks [48], as well as in the context of remote estimation use cases [49]. Under the RR policy, all nodes are scheduled in a predetermined fixed circular order. It can be argued that RR is a "good" heuristic for real-time applications as it offers fairness in time thanks to its periodic resource allocation nature. Note that the classical RR scheduler is channel-unaware by definition as it does not consider channel conditions, such as packet success probability.

2) Maximum Throughput (MT) Policy: The policy known as MT prioritizes users with the best channel quality, hence targeting throughput maximization in the network [48]. It is also known for its unfair nature as it may lead to the starvation of those users with bad channel conditions. Given our system model from section II, in which the channel quality is represented by packet loss probability, the scheduling under the MT policy can be formulated as:

$$\begin{aligned}\pi^{UL}[t] &= \operatorname*{arg\,min}_{i \in \mathcal{A}^{UL}(\boldsymbol{\Delta}[t])} p_i^{UL}[t],\\ \pi^{DL}[t] &= \operatorname*{arg\,min}_{i \in \mathcal{A}^{DL}(\boldsymbol{\Delta}[t])} p_i^{DL}[t],\end{aligned} \tag{23}$$

with:

$$\begin{aligned}\mathcal{A}^{UL}(\boldsymbol{\Delta}[t]) &\triangleq \{i:\ i \in \{1,\dots,N\}, \Delta_i^{BS}[t] > \Delta_i^{\mathcal{S}_i}[t]\}\\ &= \{1,2,\dots,N\}\\ \mathcal{A}^{DL}(\boldsymbol{\Delta}[t]) &\triangleq \{i:\ i \in \{1,\dots,N\}, \Delta_i^{\mathcal{C}_i}[t] > \Delta_i^{BS}[t]\} \cup \{\varnothing\}.\end{aligned} \tag{24}$$

Ties are broken arbitrarily. Here, $\boldsymbol{\Delta}[t] \in \mathbb{N}_0^{3N}$ denotes the AoI vector of size $3N$, where the first, second and third N indices denote, respectively, $\Delta_i^{\mathcal{S}_i}[t]$, $\Delta_i^{BS}[t]$ and $\Delta_i^{\mathcal{C}_i}[t]$ for $1 \leq i \leq N$. The reason to use $\mathcal{A}^{DL}(\boldsymbol{\Delta}[t])$ is to prevent sub-system i from being scheduled, although the BS does not have any new information to transmit. Note that this is not the case in the UL since sensors always have a new packet to transmit in our considered system model. As we are going to show later in section V, by defining feasible UL and DL actions, we are able to narrow down our search space and reduce the computational complexity of our proposed algorithm.

B. Application-Aware Greedy Policies

As an alternative to application-unaware policies, we introduce two centralized scheduling mechanisms that operate in a greedy fashion: *1) maximum age first (MAF)* and *2) maximum error first (MEF)*. Their implementation relies on the propagation of the application layer metrics such as AoI and MSE down to the data link layer; thus, they can be considered as more challenging and complex to implement when compared to RR and MT.

1) Maximum Age First (MAF) Policy: The MAF policy is a greedy strategy that schedules the user with the highest instantaneous age difference on each link [50]. That is, while the user with the highest $\Delta_i^{BS}[t]$ is selected on the UL, the DL user is obtained by subtracting $\Delta_i^{BS}[t]$ from $\Delta_i^{\mathcal{C}_i}[t]$. Hence, we formulate the MAF scheduling policy for the UL and DL as:

$$\begin{aligned}\pi^{UL}[t] &= \operatorname*{arg\,max}_{i \in \mathcal{A}^{UL}(\boldsymbol{\Delta}[t])} \Delta_i^{BS}[t],\\ \pi^{DL}[t] &= \operatorname*{arg\,max}_{i \in \mathcal{A}^{DL}(\boldsymbol{\Delta}[t])} \Delta_i^{\mathcal{C}_i}[t] - \Delta_i^{BS}[t].\end{aligned} \tag{25}$$

The key difference between MAF and RR is that MAF is adaptive to the outcome of past transmissions, whereas RR is not. In other words, if a user is scheduled in the UL but the transmission is not successful, i.e., $\pi^{UL}[t] = i, \gamma_i^{UL}[t] = 0$, one can expect that the same user is scheduled in the next time slot $t+1$ as well. However, under the RR policy, the scheduling decisions are made in a fixed order, independent of the outcome of transmissions.

[9] The control law is optimal if $\boldsymbol{x}_i = \hat{\boldsymbol{x}}$ as $\boldsymbol{L}_i^*$ minimizes F_i. Due to the impairment between the real and estimated states caused by the network, the estimation error leads to the sub-optimality of $\boldsymbol{u}$ w.r.t. the LQG cost.

2) Maximum Error First (MEF) Policy: The MEF policy is firstly proposed in [13] for single-hop multi-loop NCSs scenarios. In its original form, it prioritizes the user with the highest network-induced error, i.e., $\pi[t] = \arg\max_i ||\boldsymbol{e}_i[t]||$. However, since the actual system state $\boldsymbol{x}_i[t]$ is unknown to the centralized scheduler, it is not feasible to assume a global knowledge of $\boldsymbol{e}_i[t]$ at the scheduler. Therefore, we rule out a scheduling design based on the actual error $\boldsymbol{e}_i[t]$ and limit our study to the utilization of MSE from (20).

Moreover, we normalize the MSE by the value it takes when AoI is one. The necessity for normalization can be explained with the help of an example: Consider two control applications of different types sharing the network introduced in section II. By definition, each element of the estimation error vector $\boldsymbol{e}_i \in \mathbb{R}^{n_i}$ has the same unit as the system state $\boldsymbol{x}_i$, e.g., kelvin, meter, radians. Thus, comparing the MSE in its pure form can lead to comparison of numbers in different orders of magnitude, hence would only be possible if all sub-systems' units were identical. As a solution, we propose to normalize the MSE as:

$$nMSE_i(\Delta_i[t]) \triangleq \frac{MSE_i(\Delta_i[t])}{MSE_i(1)}, \tag{26}$$

such that the resulting metric is dimensionless. Note that this equivalent to defining c_i in (21) as $c_i \triangleq \frac{1}{MSE_i(1)}$. Throughout the paper, we employ the *normalized mean squared error (nMSE)* for scheduling instead of MSE. As a result, the policy for the MEF scheduler becomes:

$$\begin{aligned}
\pi^{UL}[t] &= \underset{i \in \mathcal{A}^{UL}(\boldsymbol{\Delta}[t])}{\arg\max} \quad nMSE_i(\Delta_i^{BS}[t]), \\
\pi^{DL}[t] &= \underset{i \in \mathcal{A}^{DL}(\boldsymbol{\Delta}[t])}{\arg\max} \quad nMSE_i(\Delta_i^{\mathcal{C}_i}[t]) - nMSE_i(\Delta_i^{BS}[t]).
\end{aligned} \tag{27}$$

The MEF scheduler characterized by (27) can be seen as a modified version of the greedy scheduler from [32], where each link can accommodate at most one transmission, and the MSE is normalized. We want to mention that MAF and MEF are channel-unaware as they are defined and implemented in this work.

IV. Optimal Joint Uplink and Downlink Finite Horizon Scheduling

In the previous section, we introduced MAF and MEF schedulers, which can be considered as heuristic solutions to the UL and DL scheduling problem for NCSs. Despite being simple in design, hence offering ease of deployment in practical scenarios, they do not guarantee optimality in task- and application-specific performance. To that end, we formulate the centralized scheduling problem as a finite horizon (FH) optimization problem and propose a policy $\boldsymbol{\Pi}_H$ that is optimal in a given time horizon H as an improvement over the greedy schedulers from section III.

Our approach is based on minimizing of the expected cost over H, which is a parameter proportional to the "farsightedness" of our scheduler. In brief, the FH optimal scheduler generates a tree structure characterized by all future state-action pairs given the current state and the channel conditions. Each state is associated with an immediate cost which is then used for the overall expected cost minimization problem. The tree structure consists of H levels incorporating all possible states and costs that could appear within H steps from now. This means that by increasing the parameter H, we also increase the time horizon that our scheduler considers while deciding on the optimal scheduling decision, thereby the computation complexity required to define the H-level tree structure.

A. States And Actions

Let $\boldsymbol{\nu}[t] \in \mathbb{N}_0^{3N}$ be a column vector of size $3N$ containing the generation time of the freshest information at sensors, BS, and controllers at time t:

$$\boldsymbol{\nu}[t] \triangleq \left[(\boldsymbol{\nu}^{\mathcal{S}}[t])^T \quad (\boldsymbol{\nu}^{BS}[t])^T \quad (\boldsymbol{\nu}^{\mathcal{C}}[t])^T\right]^T, \tag{28}$$

with:

$$\begin{aligned}
&\boldsymbol{\nu}^{\mathcal{S}}[t] \triangleq [t \quad t \quad \dots \quad t]^T, && \boldsymbol{\nu}^{\mathcal{S}}[t] \in \mathbb{N}_0^N, \\
&\boldsymbol{\nu}^{BS}[t] \triangleq \left[\nu_1^{BS}[t] \quad \nu_2^{BS}[t] \quad \dots \quad \nu_N^{BS}[t]\right]^T, && \boldsymbol{\nu}^{BS}[t] \in \mathbb{N}_0^N, \\
&\boldsymbol{\nu}^{\mathcal{C}}[t] \triangleq \left[\nu_1^{\mathcal{C}_1}[t] \quad \nu_2^{\mathcal{C}_2}[t] \quad \dots \quad \nu_N^{\mathcal{C}_N}[t]\right]^T, && \boldsymbol{\nu}^{\mathcal{C}}[t] \in \mathbb{N}_0^N.
\end{aligned} \tag{29}$$

Throughout the following analysis, we refer to the vector $\boldsymbol{\nu}[t]$ as *network state* and to a scheduling decision $\boldsymbol{\pi}[t]$ as *action*. We would like to mention that each network state $\boldsymbol{\nu}[t]$ can be mapped to a $\boldsymbol{\Delta}[t]$[10]. Nevertheless, we continue with $\boldsymbol{\nu}[t]$ throughout the following analysis for presentation purposes.

Given a network state $\boldsymbol{\nu}[t] = \boldsymbol{\nu}_\tau$ and an action $\boldsymbol{\pi}[t] = [i \quad j]$ at time t with $i \in \mathcal{A}^{UL}(\boldsymbol{\nu}[t])$, $j \in \mathcal{A}^{DL}(\boldsymbol{\nu}[t])$, we can obtain the transition probability to a next state:

$$\boldsymbol{\nu}[t+1] = f(\boldsymbol{\nu}[t], \boldsymbol{\pi}[t], \gamma_i^{UL}[t], \gamma_j^{DL}[t]), \tag{30}$$

as a function of packet loss probabilities on respective links, i.e.,:

$$\begin{aligned}
&\Pr\left[\boldsymbol{\nu}[t+1] = \boldsymbol{\nu}_{\tau+1} \;\middle|\; \boldsymbol{\nu}[t] = \boldsymbol{\nu}_\tau, \boldsymbol{\pi}[t] = [i \quad j]\right] \\
&= f(p_i^{UL}[t], p_j^{DL}[t]).
\end{aligned}$$

In particular, if $j \neq \varnothing$, there are four possible next states in our considered scenario, depending on the transmission outcome in each link. Given the network state $\boldsymbol{\nu}[t]$ as defined in (28), we can formulate the transition probabilities to the four possible next states as:

$$\begin{aligned}
\Pr[\boldsymbol{\nu}[t+1] &= \boldsymbol{\nu}_{\tau+1}^{\mathbf{11}} \mid \boldsymbol{\nu}[t] = \boldsymbol{\nu}_\tau, \boldsymbol{\pi}[t] = [i \quad j]^T] \\
&= (1 - p_i^{UL}[t]) \; (1 - p_j^{DL}[t]) \\
\Pr[\boldsymbol{\nu}[t+1] &= \boldsymbol{\nu}_{\tau+1}^{\mathbf{10}} \mid \boldsymbol{\nu}[t] = \boldsymbol{\nu}_\tau, \boldsymbol{\pi}[t] = [i \quad j]^T] \\
&= (1 - p_i^{UL}[t]) \; p_j^{DL}[t] \\
\Pr[\boldsymbol{\nu}[t+1] &= \boldsymbol{\nu}_{\tau+1}^{\mathbf{01}} \mid \boldsymbol{\nu}[t] = \boldsymbol{\nu}_\tau, \boldsymbol{\pi}[t] = [i \quad j]^T] \\
&= p_i^{UL}[t] \; (1 - p_j^{DL}[t]) \\
\Pr[\boldsymbol{\nu}[t+1] &= \boldsymbol{\nu}_{\tau+1}^{\mathbf{00}} \mid \boldsymbol{\nu}[t] = \boldsymbol{\nu}_\tau, \boldsymbol{\pi}[t] = [i \quad j]^T] \\
&= p_i^{UL}[t] \; p_j^{DL}[t]
\end{aligned} \tag{31}$$

[10] Note that, mapping a $\boldsymbol{\Delta}[t]$ to $\boldsymbol{\nu}[t]$ is not possible as a $\boldsymbol{\Delta}[t]$ does not imply a unique $\boldsymbol{\nu}[t]$. We are going to exploit this relationship between $\boldsymbol{\Delta}[t]$ and $\boldsymbol{\nu}[t]$ later to use dynamic programming for complexity reduction.

with:

$$\boldsymbol{\nu}^{\mathbf{11}}_{\tau+1} = \begin{bmatrix} \boldsymbol{\nu}^{\mathcal{S}}[t+1] \\ \nu_1^{BS}[t] \\ \vdots \\ \nu^{\mathcal{S}_i}[t] \\ \vdots \\ \nu_N^{BS}[t] \\ \nu_1^{\mathcal{C}_1}[t] \\ \vdots \\ \nu_j^{BS}[t] \\ \vdots \\ \nu_N^{\mathcal{C}_N}[t] \end{bmatrix}, \boldsymbol{\nu}^{\mathbf{10}}_{\tau+1} = \begin{bmatrix} \boldsymbol{\nu}^{\mathcal{S}}[t+1] \\ \nu_1^{BS}[t] \\ \vdots \\ \nu^{\mathcal{S}_i}[t] \\ \vdots \\ \nu_N^{BS}[t] \\ \boldsymbol{\nu}^{\mathcal{C}}[t] \end{bmatrix},$$

$$\boldsymbol{\nu}^{\mathbf{01}}_{\tau+1} = \begin{bmatrix} \boldsymbol{\nu}^{\mathcal{S}}[t+1] \\ \boldsymbol{\nu}^{BS}[t] \\ \nu_1^{\mathcal{C}_1}[t] \\ \vdots \\ \nu_j^{BS}[t] \\ \vdots \\ \nu_N^{\mathcal{C}_N}[t] \end{bmatrix}, \boldsymbol{\nu}^{\mathbf{00}}_{\tau+1} = \begin{bmatrix} \boldsymbol{\nu}^{\mathcal{S}}[t+1] \\ \boldsymbol{\nu}^{BS}[t] \\ \boldsymbol{\nu}^{\mathcal{C}}[t] \end{bmatrix}.$$

Note the differentiation between vectors written in bold, e.g., $\boldsymbol{\nu}^{BS}[t] \in \mathbb{N}_0^N$ and scalar values, e.g., $\nu_i^{BS}[t] \in \mathbb{N}_0$. Here, a 1 in the superscript indicates success in the corresponding link where the first position is the UL and the second is the DL. For instance, $\boldsymbol{\nu}^{\mathbf{10}}$ denotes the next state when UL transmission is successful, but the DL transmission fails. In the case of an empty DL transmission $j = \varnothing$, the number of next possible states reduces to two: $\boldsymbol{\nu}^{\mathbf{10}}$ and $\boldsymbol{\nu}^{\mathbf{00}}$.

B. The H-Stage Problem And Finite Horizon Cost

We consider a cost function $g : \mathbb{N}_0^{3N} \to \mathbb{R}$ mapping a network state to an immediate cost in the form:

$$g(\boldsymbol{\nu}[t]) \triangleq \sum_{i=1}^{N} g_i(\nu_i^{\mathcal{S}_i}[t], \nu_i^{BS}[t], \nu_i^{\mathcal{C}_i}[t]). \tag{32}$$

Here, $g_i : \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{R}$ characterizes the cost contribution of sub-system i. A simple example of such a function would be the weighted sum of AoI at the BS and the controller, i.e.,:

$$\begin{aligned} &g_i(\nu_i^{\mathcal{S}_i}[t], \nu_i^{BS}[t], \nu_i^{\mathcal{C}_i}[t]) \\ &= w^{\mathcal{S}_i} \underbrace{(t - \nu_i^{\mathcal{S}_i}[t])}_{=0} + w^{BS} \underbrace{(t - \nu_i^{BS}[t])}_{\Delta_i^{BS}[t]} + w^{\mathcal{C}_i} \underbrace{(t - \nu_i^{\mathcal{C}_i}[t])}_{\Delta_i^{\mathcal{C}_i}[t]}, \end{aligned}$$

with $w^{\mathcal{S}_i}, w^{BS}, w^{\mathcal{C}_i} > 0, \forall i$. By following a similar notation as in [34], we can define an additive cost of the form:

$$\mathcal{J}_t^{\boldsymbol{\Pi}_H}(\boldsymbol{\nu}[t]) \triangleq \mathop{\mathbb{E}}_{\boldsymbol{\gamma}^{UL}[\tau], \boldsymbol{\gamma}^{DL}[\tau]} \left[\sum_{\tau=t}^{t+H} g(\boldsymbol{\nu}[\tau]) \;\middle|\; \boldsymbol{\pi}[\tau] \right]. \tag{33}$$

for an initial state $\boldsymbol{\nu}[t]$ and possible future states to go, i.e., $\boldsymbol{\nu}[\tau+1] = f(\boldsymbol{\nu}[\tau], \boldsymbol{\pi}[\tau], \boldsymbol{\gamma}^{UL}[\tau], \boldsymbol{\gamma}^{DL}[\tau])$. Here, $\mathcal{J}_t^{\boldsymbol{\Pi}_H}$ corresponds to the expected H-stage cost when the scheduling policy $\boldsymbol{\Pi}_H[t] = \{\boldsymbol{\pi}[t], \boldsymbol{\pi}[t+1], \ldots, \boldsymbol{\pi}[t+H-1]\}$ is applied over the horizon H. The expectation is taken with respect to $\boldsymbol{\gamma}^{UL}[\tau]$ and $\boldsymbol{\gamma}^{DL}[\tau]$ that together with the scheduling decision $\boldsymbol{\pi}[\tau]$ define the occurrence probability of any next state. Hereby, we aim to find the optimal scheduling policy $\boldsymbol{\Pi}_H^*[t]$ for the *H-stage problem* with the optimal cost:

$$\mathcal{J}_t^{\boldsymbol{\Pi}_H^*}(\boldsymbol{\nu}[t]) = \min_{\boldsymbol{\Pi}_H} \mathcal{J}_t^{\boldsymbol{\Pi}_H}(\boldsymbol{\nu}[t]). \tag{34}$$

It is necessary to state that although $\boldsymbol{\Pi}_H$ looks H slots into the future, the optimization problem is solved repeatedly for every time slot, and the optimal action $\boldsymbol{\pi}_H^*[t]$ is taken both for UL and DL.

C. Finite Horizon Scheduler

It is shown in [34] that the H-stage problem that starts at state $\boldsymbol{\nu}[t]$ and time t, and ends at time $t+H$ can be solved optimally by minimizing the right side of equation (34) for $\tau = t+H-1, t+H-2, \ldots, t$:

$$\begin{aligned} &\mathcal{J}_\tau(\boldsymbol{\nu}[\tau]) = \\ &\min_{\boldsymbol{\pi}[\tau]} \mathbb{E}\left[g\left(\boldsymbol{\nu}[\tau]\right) + \mathcal{J}_{\tau+1}\left(f(\boldsymbol{\nu}[\tau], \boldsymbol{\pi}[\tau], \boldsymbol{\gamma}^{UL}[\tau], \boldsymbol{\gamma}^{DL}[\tau]) \right) \right], \end{aligned} \tag{35}$$

with the terminal cost:

$$\mathcal{J}_H\left(\boldsymbol{\nu}[t+H]\right) = g\left(\boldsymbol{\nu}[t+H]\right). \tag{36}$$

We refer to [34, p. 25] for proof. One can explain equations (35) and (36) in plain words as follows: the optimal cost $\mathcal{J}^{\boldsymbol{\Pi}_H^*}(\boldsymbol{\nu}[t])$ can be obtained by iterating backwards in time from stage $H-1$ to stage 0, while at each iteration step τ the optimal action $\boldsymbol{\pi}^*[\tau]$ solving (35) is taken. Consequently, the scheduling policy $\boldsymbol{\Pi}_H^*[t] = \{\boldsymbol{\pi}^*[t], \boldsymbol{\pi}^*[t+1], \ldots, \boldsymbol{\pi}^*[t+H-1]\}$ achieving the minimum expected cost $\mathcal{J}_t^{\boldsymbol{\Pi}_H^*}(\boldsymbol{\nu}[t])$ is the optimal joint UL and DL scheduling policy.

The H-stage problem can be modeled as a tree structure with H levels where each network state is represented as a node. The 0-th level, also called *root*, consists of a single node with the state $\boldsymbol{\nu}[t]$. Given the root node, the remaining tree is defined by all possible state-action pairs governed by the joint UL and DL scheduling problem. In particular, an action $\boldsymbol{\pi}[\tau]$ given a network state $\boldsymbol{\nu}[\tau]$ implies up to 4 possible next states, as described in (31), where each transition is modeled by an outgoing edge from $\boldsymbol{\nu}[\tau]$. Moreover, each edge is assigned a transition probability depending on the action, i.e., scheduling decision, and the transmission success probability of the respective links. Once the whole tree is generated, each state's cost is assigned according to (35) starting from the last level of the tree, which is also the H-th level. The nodes in the last level are called *leaf nodes*. The following steps summarize the resulting algorithm of the FH scheduler:

Finite Horizon Scheduling Algorithm

1. Initialize the current state $\boldsymbol{\nu}[t]$ as the root of the tree.
2. Starting from the root, determine the feasible actions $\boldsymbol{\pi}[\tau]$ according to $\mathcal{A}^{UL}(\boldsymbol{\nu}[\tau])$ and $\mathcal{A}^{DL}(\boldsymbol{\nu}[\tau])$ and subsequently all possible next states $\boldsymbol{\nu}(\tau+1)$ given an action $\boldsymbol{\pi}[\tau]$.
3. Add the obtained next states as child nodes to the next level of the tree and initialize the edges between the parent and child nodes with corresponding transition probabilities.
4. Repeat steps 2. and 3. until the H-th level of the tree is constructed.
5. Assign the minimum cost and best action to each node as in (35) starting from the leaf nodes.

Once the FH scheduling algorithm completes, the scheduler executes the optimal action $\boldsymbol{\pi}^*_H[t]$ assigned to the root node. We emphasize that even though the FH scheduling algorithm has obtained the optimal action for all possible future states within the horizon H, the FH scheduler algorithm must be repeated after every time slot. The reason is the modification of the tree's root node and the dynamically changing edge transition probabilities caused by time-varying channel conditions. We would like to remind the reader that the scheduler knows each link's current loss probability but is unaware of future channel conditions, e.g., $p_i^{UL}[t+1], p_i^{DL}[t+2], \ldots$. Therefore, we use Chapman-Kolmogorov equations to obtain the expected reliability of each link quality in $t+h$ given the current channel state [51, Ch. 4.2][11].

Looking at the algorithm, one can easily deduct that constructing such a tree structure is a heavy task in terms of computational complexity. In particular, the complexity strictly depends on the chosen farsightedness, i.e., H and the network size, i.e., N. A detailed discussion on the FH algorithm's complexity can be found later in section V-C2.

Selecting the cost function: We propose to employ an additive weighted cost function for immediate state cost given as:

$$g(\boldsymbol{\nu}[t]) = \sum_{i=1}^{N} w^{BS} \underbrace{nMSE_i(\Delta_i^{BS}[t])}_{\triangleq g_i^{BS}[t]} + w^{\mathcal{C}_i} \underbrace{nMSE_i(\Delta_i^{\mathcal{C}_i}[t])}_{\triangleq g_i^{\mathcal{C}_i}[t]}, \tag{37}$$

where $w^{BS}, w^{\mathcal{C}_i} > 0, \forall i$. $\Delta_i^{BS}[t]$ and $\Delta_i^{\mathcal{C}_i}[t]$ are defined as in (3), (10), respectively. Simply put, (37) considers not only the nMSE at controllers but also the nMSE at the BS. The reason behind the consideration of $\Delta_i^{BS}[t]$ can easily be explained with the help of a toy example as follows: Suppose a newly initialized network with $N=1$ and $H=1$. Moreover, assume $\Delta_1^{\mathcal{C}_i}[0] = \Delta_1^{BS}[0] = 5$ and $\Delta_i^{\mathcal{S}_i}[0] = 0$. If we only reward a cost reduction at the controller, or equivalently, if $w^{BS} = 0$ but $w^{\mathcal{C}_1} > 0$, a successful UL transmission in time slot t does not lead to any change in cost within the considered horizon, i.e., $t+1$. As a result, the scheduler is indifferent between $\pi^{UL}[t] = 1$ and $\pi^{UL}[t] = \varnothing$ as the network is not better-off by scheduling any sub-system within the considered finite horizon. Therefore, to incorporate a hidden future reward enabled through an update of the BS into our decision-making, we choose a positive w_{BS} in our framework.

One may use different cost functions than nMSE as it has been done in the existing literature. Through our simulation results, we will show later in section V that the FH scheduling algorithm can operate with various age-penalty functions. For instance, the scheduling algorithm can employ $g_i^{BS/\mathcal{C}_i}[t] = \Delta_i^{BS/\mathcal{C}_i}[t]$ to improve information freshness by solving the H-stage problem. Similarly, common age-penalty functions such as $g_i^{BS/\mathcal{C}_i}[t] = e^{\alpha_i \Delta_i^{BS/\mathcal{C}_i}[t]}$ and $g_i^{BS/\mathcal{C}_i}[t] = \alpha_i \Delta_i^{BS/\mathcal{C}_i[t]}$ with a design parameter $\alpha_i > 0$ as a multiplier constant in front of AoI are other examples of such. Throughout the following section, we simplify the notation by dropping the superscript. To name an example, when we say that the scheduler employs the metric $\Delta_i[t]$, it implies $g_i^{BS}[t] = \Delta_i^{BS}$ and $g_i^{\mathcal{C}_i}[t] = \Delta_i^{\mathcal{C}_i}[t]$.

A greedy equivalent of H=1: The FH scheduling algorithm describes a Monte Carlo tree search (MCTS) algorithm with states and actions being governed by the system model described in section II. On the one hand, one can argue that this brings an additional effort for the system designer to implement such a framework in software. On the other, the MEF scheduler, which has been proposed in the preliminary version of this work, neglects the reliability of each link by definition, which may cause severe performance loss in practice. To that end, we propose a channel-aware alternative to the MEF policy that is in fact an equivalent of the FH scheduling algorithm when H is equal to one.

First, we notice that for $H=1$, the FH algorithm divides the UL and DL scheduling problem into two independent problems[12]. Next, let us look at the UL case with the network state $\boldsymbol{\nu}[t]$ and $g_i^{BS}[t]$ as in (37). Suppose none of the users are scheduled in slot t, implying $g_i^{BS}[t+1] = nMSE_i(\Delta_i^{BS}[t]+1)$ for all i. However, for $\pi^{UL}[t] = i$, the BS cost to be incurred, i.e., $g_i^{BS}[t+1]$, becomes $nMSE_i(1)$ and $nMSE_i(\Delta_i^{BS}[t]+1)$ with probabilities $1 - p_i^{UL}[t]$ and $p_i^{UL}[t]$, respectively, depending on the transmission outcome. Hence, we can express the total expected BS cost at $t+1$ for $\pi^{UL}[t] = i$:

$$\mathbb{E}\left[\sum_{i=1}^{N} g_i^{BS}[t+1] \;\middle|\; \pi^{UL}[t] = i\right] = \\ \sum_{j=1}^{N} \left(nMSE_j(\Delta_j^{BS}[t]+1)\right) - nMSE_i(\Delta_i^{BS}[t]+1) + \\ (1 - p_i^{UL}[t])nMSE_i(1) + p_i^{UL}[t]nMSE_i(\Delta_i^{BS}[t]+1). \tag{38}$$

As a result, the optimal policy for the UL can be obtained by

[11] An alternative would be to assume that the current channel conditions would remain constant throughout the finite horizon, similar to [33].

[12] Since we have one slot transmission delay on each link, an UL decision considers only a possible cost reduction at the BS and is myopic to any future costs.

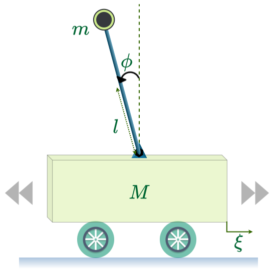

Fig. 4: An inverted pendulum with motorized cart. The primary goal of the controller is to hold the pendulum in upright position, i.e., to keep $|\phi|$ close to zero as much as possible.

minimizing the RHS of (38), or, equivalently:

$$\max_i \quad \big\{(1-p_i^{UL}[t])\, nMSE_i(\Delta_i^{BS}[t]+1) - (1-p_i^{UL}[t])\, nMSE_i(1)\big\}. \tag{39}$$

A similar analysis can be done for the DL with the replacing the cost at the controller after a failed and successful DL transmission accordingly. That is, $g_i^{\mathcal{C}_i}[t+1] = nMSE_i(\Delta_i^{\mathcal{C}_i}[t]+1)$ for the success case and $g_i^{\mathcal{C}_i}[t+1] = nMSE_i(\Delta_i^{BS}[t]+1)$ otherwise. As a result, the DL equivalent of the FH algorithm with $H=1$ becomes:

$$\max_i \quad \big\{(1-p_i^{DL}[t])\, nMSE_i(\Delta_i^{\mathcal{C}_i}[t]+1) - (1-p_i^{DL}[t])\, nMSE_i(\Delta_i^{BS}[t]+1)\big\}. \tag{40}$$

The resulting policy can be interpreted as the prioritization of the sub-system with the *maximum expected error reduction (MEER)* on each link. Note that although the MEER is an analytical equivalent of the FH algorithm for $H=1$, it possesses three very important properties. Firstly, it solves the channel-unawareness weakness of the MEF algorithm by introducing the reliability of each link into the equation. Secondly, it has a computational complexity of $\mathcal{O}(N)$ per slot, hence, is scalable. Lastly, and most importantly, it does not require the implementation of a MCTS algorithm, thus, mitigating the implementation effort for practical deployment.

V. Simulation Results and Discussion

In this section, we evaluate the performance of our proposed FH scheduler in terms of average AoI, capturing information freshness, average nMSE capturing estimation performance, and average LQG cost quantifying the control performance in the network. Our evaluation is not only limited to the presentation of key performance indicators (KPIs) when various selected cost functions from the existing literature are employed, but additionally, we investigate the effect of increasing the FH parameter H on the resulting performance.

A. Simulation Details

Our simulation setup consists of $N=4$ feedback control loops, each belonging to a different type. In particular, one of these sub-systems is an inverted pendulum (IP), which is a well-known real-life application in control theory textbooks [43]. As depicted in Fig. 4, it consists of a pendulum mounted on a motorized cart where the controller's objective is to hold the pendulum in an upright position by moving the cart back and forward. While the IP has a four-by-four system matrix, i.e., $\boldsymbol{A}_{IP} \in \mathbb{R}^{4\times 4}$, the remaining applications are each a scalar control loop with $\boldsymbol{A}_{i\neq IP} \in \mathbb{R}^{1\times 1}$. The parameter selection of our control model can be summarized as:

$$\boldsymbol{A}_2 = \boldsymbol{A}_{IP} = \begin{bmatrix} 1.0000 & 0.0100 & 0.0001 & 0.0000 \\ 0.0000 & 0.9983 & 0.0191 & 0.0001 \\ 0.0000 & 0.0000 & 1.0017 & 0.0100 \\ 0.0000 & -0.0049 & 0.3351 & 1.0017 \end{bmatrix},$$

$$\boldsymbol{B}_2 = \boldsymbol{B}_{IP} = \begin{bmatrix} 0.0001 & 0.1706 & 0.0002 & 0.0488 \end{bmatrix}^T,$$

$$\boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_{IP} = \begin{bmatrix} 6.4\ 10^{-7} & 0 & 0 & 0 \\ 0 & 4.9\ 10^{-7} & 0 & 0 \\ 0 & 0 & 2.74\ 10^{-5} & 0 \\ 0 & 0 & 0 & 4.87\ 10^{-5} \end{bmatrix},$$

$$\boldsymbol{A}_1 = \begin{bmatrix} 1.0 \end{bmatrix}, \boldsymbol{A}_3 = \begin{bmatrix} 1.2 \end{bmatrix}, \boldsymbol{A}_4 = \begin{bmatrix} 1.3 \end{bmatrix},$$

$$\boldsymbol{B}_1 = \boldsymbol{B}_3 = \boldsymbol{B}_4 = \begin{bmatrix} 1.0 \end{bmatrix}, \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_3 = \boldsymbol{\Sigma}_4 = \begin{bmatrix} 1.0 \end{bmatrix}.$$

Moreover, the selected parameters to determine the stabilizing feedback gain are given as:

$$\boldsymbol{Q}_1 = \boldsymbol{Q}_3 = \boldsymbol{Q}_4 = \begin{bmatrix} 100.0 \end{bmatrix}, \boldsymbol{Q}_2 = \begin{bmatrix} 5000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{R}_1 = \boldsymbol{R}_2 = \boldsymbol{R}_3 = \boldsymbol{R}_4 = \begin{bmatrix} 1.0 \end{bmatrix}.$$

Our results consist of 50 simulation runs that are each $T = 180\,000$ time slots long. The finite horizon H is varied from 0 to 5, which adjusts the farsightedness of the scheduling policy. When $H=0$ is selected, we employ the greedy MAF policy described in section III-A. Similarly, the greedy MEF scheduler corresponds to a combination of $H=0$ with the metric $nMSE_i(\Delta_i[t])$.

For the cost function from (37), we select equal contribution by the intermediate and destination nodes, i.e., $w^{BS} = w^{\mathcal{C}_i} = 1, \forall i$. Additionally, to model the time-varying channel quality on UL and DL, we utilize the Gilbert-Elliott model with the following parameters: $p_G = 0.1$, $p_B = 0.4$, $p_{G2B}^{UL} = p_{G2B}^{DL} = p_{B2G}^{UL} = p_{B2G}^{DL} = 0.1$.

Fig. 5 illustrates the evolution of nMSE with increasing AoI for multiple sub-systems considered in our simulations. As depicted in the figure, each sub-system has a different expected nMSE given the instantaneous age. For instance, for $\Delta = 7$, the nMSe is expected to be much higher for $\boldsymbol{A}_4$ than for $\boldsymbol{A}_1$, indicating a higher estimation error normalized by their respective default state $\Delta = 1$. In addition, we observe that the IP lies between $\boldsymbol{A}_1$ and $\boldsymbol{A}_3$ w.r.t. nMSE. In fact, it is comparable to an imaginary[13] scalar sub-system with $\boldsymbol{A} = [1.1]$ and $\boldsymbol{\Sigma} = [1.0]$. This brings us to the selection α_i in our simulations and in the following discussion. That is, we select $\alpha_i = \boldsymbol{A}_i$ for $i \in \{1,3,4\}$ and $\alpha_2 = \alpha_{IP} = 1.1$ when applicable. We would like to stress that α_i appears as a multiplicative factor in front of AoI for some of the considered cost metrics, such as $g_i^{BS/\mathcal{C}_i}[t] = \alpha_i \Delta_i^{BS/\mathcal{C}_i}[t]$.

[13] By using the word "imaginary", our goal is to emphasize that this sub-system is not considered in our simulations. It is shown in the figure only for comparison.

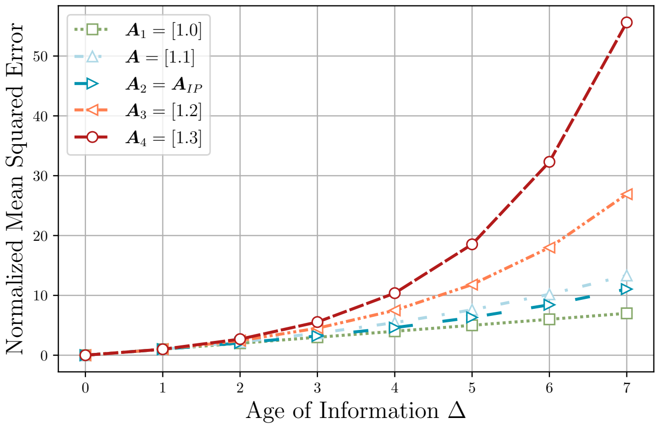

Fig. 5: The evolution of the normalized mean squared error with increasing AoI. For scalar systems, i.e., $i \in \{1, 3, 4\}$, as the system matrix $\boldsymbol{A}_i$ increases, a given Δ leads to a higher squared error in expectation. Additionally, the nMSE increase of IP in AoI is comparable to an imaginary scalar sub-system with $\boldsymbol{A}_i = [1.1]$ and $\boldsymbol{\Sigma}_i = [1.0]$.

B. Key Performance Indicators (KPIs)

We select the average AoI as the primary KPI to capture information freshness in the network[14]. The average AoI during a simulation run is measured per time slot and user; hence it can be obtained by:

$$\overline{\Delta} = \frac{1}{T}\frac{1}{N}\sum_{t=0}^{T-1}\sum_{i=1}^{N}\Delta_i^{\mathcal{C}_i}[t]. \tag{41}$$

Our goal is not to minimize $\overline{\Delta}$ but to minimize the control-related KPIs. Therefore we present the two main metrics for the following evaluation, i.e., normalized MSE and LQG cost are calculated per time slot and per user as:

$$\overline{nMSE} = \frac{1}{T}\frac{1}{N}\sum_{t=0}^{T-1}\sum_{i=1}^{N} w_i \; nMSE_i(\Delta_i[t]), \tag{42}$$

$$\overline{\mathcal{F}} = \frac{1}{N}\sum_{i=1}^{N} w_i \; \mathcal{F}_i, \tag{43}$$

with $\mathcal{F}_i$ as in (14) and sub-system weight factor $w_i \geq 0, \forall i$. In this work, we select $w_i = 1$ for all sub-systems. Note that a lower $\overline{nMSE}$ indicates a better estimation performance, whereas a lower $\overline{\mathcal{F}}$ points to increased control performance in the network.

C. Simulation Results and Discussion

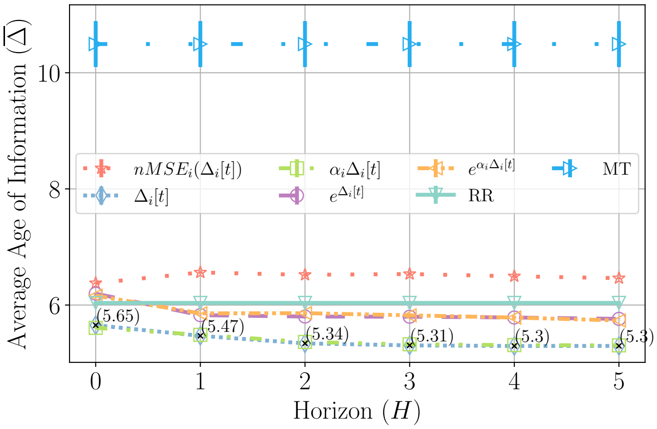

Fig. 6: Average age of information (AoI), i.e., $\overline{\Delta}$, in the network. While maximum throughput (MT) scheduler performs the worst w.r.t. $\overline{\Delta}$, using linear age-penalty functions, i.e., $\alpha_i\Delta_i[t]$, $\Delta_i[t]$ leads to the lowest average AoI in the network. Vertical error bars represent 95% confidence interval.

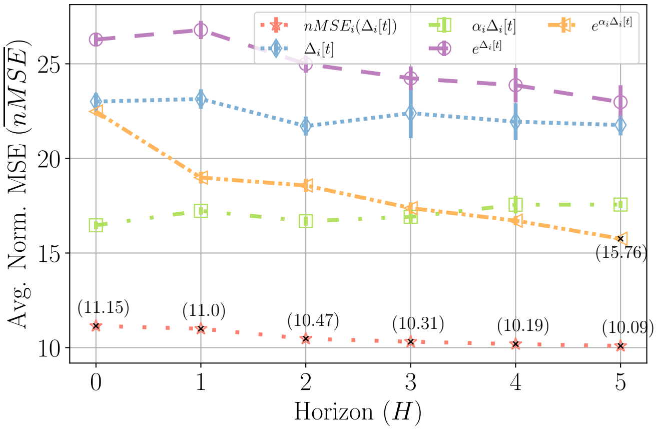

Fig. 7: Average normalized mean squared error (nMSE) in the network. In the x-axis, the horizon parameter $0 \leq H \leq 5$ is varied. The nMSe from (42) is plotted in the y-axis. Vertical error bars represent 95% confidence interval.

Let us first compare the resulting information freshness performance of the considered scheduling policies presented in Fig. 6. The figure plots $\overline{\Delta}$ against increasing H from zero to five, where each line represents a different scheduling policy, including MT, RR, and FH scheduler operating with various age-penalty functions. As evident from the figure, the best-performing schedulers in terms of $\overline{\Delta}$ are the FH scheduler with linear age-penalty functions, i.e., $\Delta_i[t]$ and $\alpha_i\Delta_i[t]$. In addition, we can conclude that the MT scheduler is not suitable for providing fresh information as it prioritizes those users with the best channel conditions, irrespective of information semantics. On the contrary, although being a channel-unaware policy, RR performs significantly better than MT due to its periodic scheduling pattern. The main contribution of this work, i.e., the FH scheduler using $nMSE_i[t]$ as the scheduling metric, falls behind RR when it comes to providing fresh information. Please note that the results for MT and RR are included multiple times in the figure for presentation purposes, although they are independent of the horizon parameter H. Thus, they appear as a horizontal line over the entire figure.

As our primary goal is not to provide fresh information but to help users perform better at their underlying communication

[14] Performing better or worse in information freshness is not a success indicator for quality of control. Average AoI is presented solely for more insightful discussion.

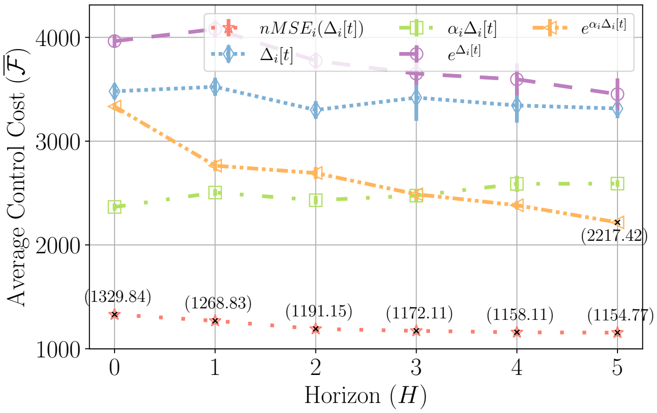

Fig. 8: Average control cost in the network. The control cost $\mathcal{F}_i$ per sub-system is defined as in (14). Vertical error bars represent 95% confidence interval. The selection of α_i is discussed in section V-A.

purpose, we examine figures 7 and 8 displaying the estimation and control performances of the considered schedulers. Fig. 7 shows the average estimation performance for varying H captured by $\overline{nMSE}$ from (42). For presentation purposes, we intentionally left RR and MT out as they perform significantly worse than any other FH schedulers considered in this work[15]. Looking at the figure, we can easily see that the FH scheduler with $nMSE_i(\Delta_i[t])$ leads to the best estimation performance in the network. This is an expected result since the scheduling metric used for decision-making is directly related to the KPI, and the FH scheduler offers optimality for the selected horizon. In particular, applying the MEF scheduler from [32], which coincides with the greedy version of our proposed policy, i.e., $H = 0$, outperforms its closest competitor, i.e., the combination of $H = 5$ with $e^{\alpha_i \Delta_i[t]}$, by more than 29%. Moreover, by increasing the farsightedness of our scheduler beyond $H = 0$, we are able to improve the estimation performance further by 9.5%. It is important to mention that the approximating functions $\alpha_i \Delta_i$ and $e^{\alpha_i \Delta_i[t]}$ outperform the FH age-optimal $\Delta_i[t]$ scheduler. This reveals the potential in approximating the task criticality by appropriate functions for those scenarios where the exact modeling is not feasible.

Fig. 8 illustrates the average control cost $\overline{\mathcal{F}}$ in the network for increasing H. As in the case of $\overline{nMSE}$, the best-performing scheduling policy in terms of control cost is our proposed FH scheduler using $nMSE_i(\Delta_i[t])$. In particular, when $H = 5$ is selected, the achieved $\overline{\mathcal{F}}$ is lower than the greedy scheduler proposed in [32] by more than 13%, which indicates an improved control performance. Similar to Fig. 7, we observe that the utilization of homogeneous metrics for all sub-systems, as in the case of $\Delta_i[t]$ and $e^{\Delta_i[t]}$, falls behind carefully selected heterogeneous cost functions, i.e., $\alpha_i \Delta_i$, $e^{\alpha_i \Delta_i[t]}$, $nMSE_i(\Delta_i[t])$. Notice that none of the selected schedulers incorporate control cost into their decision-making. However, as in the case of our proposed FH scheduler, the control cost can be indirectly improved as a side-product of estimation error minimization.

	MT	RR	FH
I)	0/50	41/50	50/50
II)	5/50	50/50	50/50
III)	7/50	50/50	50/50

TABLE I: Number of simulation runs out of 50, in which the pendulum angle satisfied the success criteria I, II, and III.

We would like to emphasize that the amount of available network resources has been kept constant throughout our simulations. Only by changing how the limited network resources are distributed among demanding applications, we are able to create a significant performance gap between multiple scheduling strategies. Therefore, we can clearly observe and state that by introducing task-awareness into the network's decision-making, e.g., into the MAC layer protocols, one can significantly vary the quality of the offered service. Both figures presenting the achieved $\overline{nMSE}$ and $\overline{\mathcal{F}}$ confirm the importance of such an approach and reveal the potential of STOC as an alternative to increasing the bandwidth to offer the same level of service.

Remarks on inverted pendulum's performance: In order to demonstrate the advantage of control-aware network design, e.g., our proposed FH scheduler, over conventional methods, e.g., RR and MT, in a more tangible way, let us focus specifically on our selected real-life application, i.e., IP depicted in Fig. 4. Let us define three success criteria specific for the maximum allowed pendulum deviation angle $|\phi[t]|$ in degrees as:

I) $|\phi[t]| < 15°$ for $0 \leq t < T$,
II) $|\phi[t]| < 30°$ for $0 \leq t < T$,
III) $|\phi[t]| < 90°$ for $0 \leq t < T$.

We are interested in the number of measurement runs out of 50, in which the maximum pendulum angle deviation is less than the selected upper bound. Table I summarizes the number of successful runs for RR, MT, and FH scheduler with $H = 5$ utilizing $nMSE_i[t]$.

In simple words, the FH scheduler is able to keep the pendulum angle within ±15° throughout our 50 simulation runs. On the other hand, when we select MT scheduler, only 7 out of 50 runs comply with criteria III. This means that a human observer, looking at the IP depicted in Fig. 4, would see the pendulum falling in 43 runs as the pendulum angle exceeds ±90°. Moreover, RR leads to satisfactory results in all runs only w.r.t. criteria II and III.

1) Increasing the network size: So far, we have only considered $N = 4$ sub-systems. In order to show the effectiveness of our algorithm in a larger network, let us increase N further and vary it from 6 to 12 in steps of two[16]. Fig. 10 shows the resulting performance w.r.t. nMSE. While the boxplots correspond to the FH algorithm, the horizontal lines present the $n\bar{M}SE$ achieved by the MEER scheduler. For instance, when $N = 12$, the FH scheduler with $H = 3$ improves the performance by more than 10% compared to $H = 0$. Note that

[15] The average nMSE for RR and MT are $12.9 \cdot 10^4$ and $34.4 \cdot 10^{10}$, respectively. The large numbers are mainly caused by more unstable sub-systems with $\boldsymbol{A}_i \in \{1.2, 1.3\}$ reaching high age values.

[16] For each configuration, we select half of the sub-systems as inverted pendulums and the other half as $A_i = 1.0$.

H	1	2	3	4	5
WC	26	651	16276	4.1 10^5	10.17 10^6
Sim.	16.3	238.6	3428	4.9 10^4	7.26 10^5

TABLE II: Average node count in the Monte Carlo Tree Search (MCTS) for varying H.

we select a smaller H for a larger N to reduce the complexity. Nevertheless, the results confirm that our proposed algorithm works for various settings and not only for a specific one.

2) Discussion on complexity: The selection of the FH parameter H is crucial for our proposed FH scheduling policy. By increasing H, we are able to control the time horizon, in which the FH scheduler offers optimality for the selected metric. However, this comes with an exponential complexity increase for the joint UL and DL scheduling problem. Particularly, the maximum number of states defining the H-stage problem for N sub-systems is formulated by the following equation:

$$\#nodes(H) = \frac{(N^2 + 2N + 1)^{H+1} - 1}{N^2 + 2N}. \tag{44}$$

In other words, assuming that N is given, the H parameter allows us to control up to how many nodes the tree structure from section IV-C should consist of. Note that when the greedy policy is selected, i.e., $H = 0$, the tree consists of a single root node representing the current network state. Table II presents the number of nodes in the worst-case (WC) obtained by (44) and the average node count measured during our simulations. The difference between WC and simulation is caused by limiting the set of possible DL actions to only those subsystems that have a packet to be transmitted in the DL, i.e., $\mathcal{A}^{DL}$, through which we are able to narrow down our search space.

The computational complexity grows exponentially for increasing H. However, the performance gain that we are able to achieve by increasing the FH does not grow at the same speed. That is, a significant portion of the gain is achieved already when H is increased up to three in our considered scenario. As we go beyond $H = 3$, the benefit of looking into a longer time horizon diminishes. Therefore, the right H value should be identified prior to deployment if the complexity constitutes a bottleneck for the considered scenario.

It is important to emphasize once again that selecting a very large H is not our primary goal. In fact, the dynamically changing channel conditions render the solution to a very large H not attractive. To elaborate, if we choose a very large H, we would not only be increasing the complexity of the considered problem, but instead we would solve an inaccurate problem, due to the mismatch between the transition probabilities in the decision tree and the system's actual evolution. An alternative approach to the FH algorithm is to assume an average link reliability, p_i, that is time-invariant[17]. Hence, one could formulate an infinite horizon problem and solve it optimally using the value iteration algorithm [52]. However, this would imply that the scheduler is independent of the instantaneous channel conditions, hence making a sub-optimal scheduling decision based on p_i, instead of $p_i[t]$.

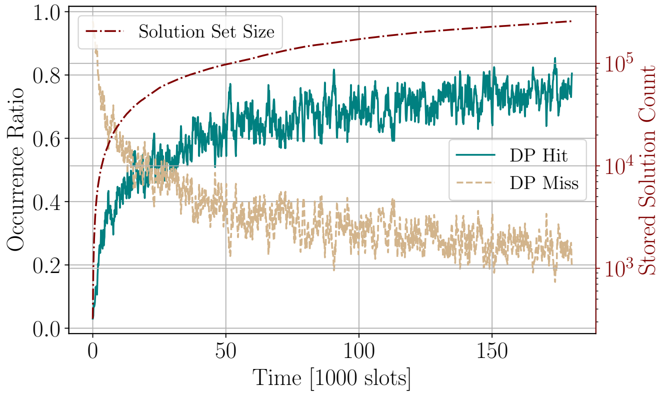

Fig. 9: The figure shows the ratio of a problem being already in the solution set by using dynamic programming method (DP) and the occurrence ratio of a new problem. Moreover, the size of the solution set is plotted in logarithmic scale.

3) Tackling complexity in practical scenarios: There are a few techniques one could employ to combat complexity issues in practical deployment.

Learning Optimal Action(s): In a static network with an invariant number of control loops, the FH scheduler can *learn* the optimal actions over time as the same states are revisited over and over. That is, if the same H-stage problem has been solved and the optimal action(s) has been determined before, it can be stored in the memory for future reference. This approach is similar to the dynamic programming (DP) method used in computer science, which stores solutions to sub-problems for future usage.

One way to speed up the learning process is to use parallel computing. In fact, our simulations employ the DP approach by running multiple threads that collectively gather a solution set in the shared memory. To demonstrate the potential of this approach, we present Fig. 9. The figure shows the ratio of an H-stage problem being already in the solution set, which we call a *DP hit*. If the H-stage problem is identified as novel, referred to as a *DP miss*, the optimal solution is calculated and added to the solution set afterwards. As we see from the figure, as the network learns more solutions over time, the occurrence ratio of a DP hit takes over, and a new execution of the FH scheduling algorithm is not needed. It is needless to say that storing previous solutions to already visited problems comes with an increased memory cost over time. In brief, one can say that the high computational complexity is traded for increased memory demand.

As an important remark, we would like to mention that we are able to identify a problem reappearing due to the limited variance in the link quality, thanks to the Gilbert-Elliott model. That is, users experience a dynamic packet loss probability on each link that alternates between two values, i.e., p_G and p_B depending on the current state, i.e., G and B. However, each of these probabilities would take a real value between zero and one in a real network. Consequently, the H-stage problem is highly likely to be a novel problem, leading to a DP miss. Nev-

[17] For instance, if the channel is modeled by the GE model, one could calculate each link's long-term average loss probability.

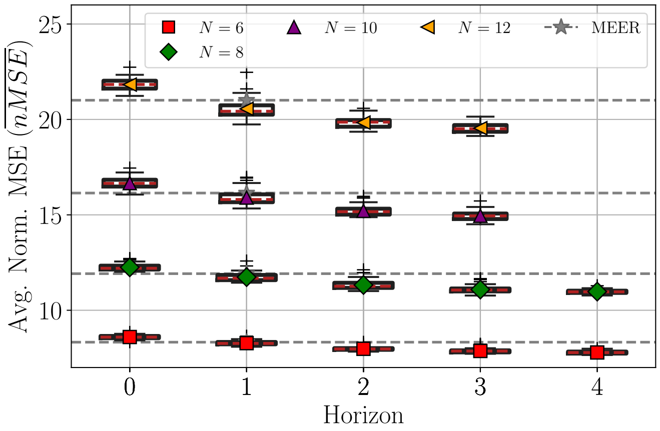

Fig. 10: Detailed average normalized mean squared error (nMSE) results when then number of sub-systems, i.e., N is varied. The markers correspond to means. The horizontal line corresponds to the MEER scheduler.

ertheless, this issue motivates the consideration of approaches similar to [38]. In [38], authors adjust the parameters of a two-state GE model through adapting the coefficient of variation identified in the real network trace. They demonstrate that the model they obtain after the parametrization step captures the loss pattern of the real network fairly well.

Gradual increase in H: The main benefit of utilizing the FH algorithm over greedy alternatives is that it enables us to control the complexity of the problem through H. This can be a very useful property, if combined with the dynamic learning presented above. More specifically, the centralized scheduler can start with a low complexity version, e.g., $H = 1$, and learn the optimal policy. To improve the performance further, the scheduler can revisit some of the previously solved problems and modify the optimal policy (if necessary) by solving it for a larger H. A trivial way to determine which states to revisit could be for instance comparing the occurrence frequency of each state and choosing the highest ones.

4) Simulation versus analytic methods: As a replacement of a comparison based on Monte Carlo experiments, one could argue for finding an analytical solution to the long-term average nMSE minimization problem characterized in (42). To name an example, if one is provided with the stationary distribution of AoI given a stationary scheduling policy $\mathbf{\Pi}$, it can be mapped to an average nMSE performance. However, such an approach provides only partial answers, e.g., unclear mapping between AoI distribution and average LQG cost from (43). Additionally, such a methodology comes with a significant difficulty without contributing substantially to the considered problem. On the other hand, a simulation-based study, as we have conducted in this work, gives definitive answers to the KPIs of interest. Furthermore, it serves as a proof that our proposed solution is practically implementable.

VI. Conclusion And Future Work

Networked control systems (NCSs) belong to the category of task-oriented communications, which are feedback control loops closed over a network. Their communication purpose is to drive the system state to the desired value through the controller's inputs when there is at least one non-ideal link in the feedback loop. As their control performance is tightly coupled with the service provided by the communication, a possible way to minimize the performance degradation caused by imperfect communication is optimal decision-making through joint consideration of network and control.

In this work, we investigate the joint uplink and downlink scheduling problem for feedback control loops closed over a star network. The medium access on each link is managed by a centralized scheduler that is responsible for allocating the wireless network resources. The control subsystems experience time-varying packet loss probability on each link where a two-state Gilbert-Elliott model characterizes the link quality. As the main contribution, we propose a finite horizon (FH) scheduler that is optimal in a given time horizon H and compare the performance it achieves w.r.t. control-related key performance indicators (KPIs) to other scheduling policies from the literature. Through simulations, we show that our proposed FH scheduler is able to outperform all other considered schedulers when it comes to control task-specific KPIs. This work is an extension of our previous work [33], in which we consider a single-hop network. Additionally, different than [32], this work considers a broader range of age-penalty functions from the literature that have been utilized for decision-making in the network.

One of the possible extensions of this work would be the implementation of the FH scheduling algorithm on real hardware, e.g., using software-defined radios as in [47]. Moreover, the proposed theoretical framework can be employed to capture alternative task-criticalities, contexts, and semantics through the replacement of the utilized scheduling metric, i.e., nMSE, with a different age-penalty function. Last but not least, one could study the FH scheduling algorithm's performance for highly mobile scenarios, in which the channel conditions change more rapidly and drastically.

References

[1] W. Saad, M. Bennis, and M. Chen, "A vision of 6g wireless systems: Applications, trends, technologies, and open research problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.

[2] E. Calvanese Strinati and S. Barbarossa, "6G networks: Beyond Shannon towards semantic and goal-oriented communications," *Computer Networks*, vol. 190, 2021.

[3] E. Uysal, O. Kaya, A. Ephremides, J. Gross, M. Codreanu, P. Popovski, M. Assaad, G. Liva, A. Munari, B. Soret, T. Soleymani, and K. H. Johansson, "Semantic communications in networked systems: A data significance perspective," *IEEE Network*, 2022.

[4] J. Nilson, "Real-time control systems with delays," *PhD Dissertation*, 1998, dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden.

[5] A. Cervin, "Integrated control and real-time scheduling," *PhD Dissertation*, 2003, lund Institute of Technology, Lund, Sweden.

[6] M. Drew, X. Liu, A. Goldsmith, and K. Hedrick, "Networked control system design over a wireless lan," in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 6704–6709.

[7] K. Liu, E. Fridman, and K. H. Johansson, "Networked control with stochastic scheduling," *IEEE Transactions on Automatic Control*, vol. 60, no. 11, pp. 3071–3076, 2015.

[8] D. Maity, M. H. Mamduhi, S. Hirche, and K. H. Johansson, "Optimal lqg control of networked systems under traffic-correlated delay and dropout," *IEEE Control Systems Letters*, vol. 6, pp. 1280–1285, 2022.

[9] X. Liu and A. Goldsmith, "Wireless network design for distributed control," in *IEEE Conference on Decision and Control (CDC)*, vol. 3, 2004, pp. 2823–2829.

[10] K. Gatsis, A. Ribeiro, and G. J. Pappas, "Decentralized channel access for wireless control systems," *IFAC-PapersOnLine*, vol. 48, no. 22, pp. 209–214, 2015.

[11] A. Molin and S. Hirche, "Price-based adaptive scheduling in multi-loop control systems with resource constraints," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3282–3295, 2014.

[12] M. Vilgelm, M. H. Mamduhi, W. Kellerer, and S. Hirche, "Adaptive decentralized mac for event-triggered networked control systems," in *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '16, 2016, p. 165–174.

[13] G. Walsh and H. Ye, "Scheduling of networked control systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 57–65, 2001.

[14] G. Walsh, H. Ye, and L. Bushnell, "Stability analysis of networked control systems," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 3, pp. 438–446, 2002.

[15] A. Molin, H. Esen, and K. H. Johansson, "Scheduling networked state estimators based on value of information," *Automatica*, vol. 110, 2019.

[16] X. Zheng, S. Zhou, and Z. Niu, "Urgency of information for context-aware timely status updates in remote control systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7237–7250, 2020.

[17] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2011, pp. 350–358.

[18] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *IEEE International Conference on Computer Communications (INFOCOM)*, 2012, pp. 2731–2735.

[19] B. Buyukates, A. Soysal, and S. Ulukus, "Age of information in multihop multicast networks," *Journal of Communications and Networks*, vol. 21, no. 3, pp. 256–267, 2019.

[20] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Scheduling policies for minimizing age of information in broadcast wireless networks," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2637–2650, 2018.

[21] T. Shreedhar, S. K. Kaul, and R. D. Yates, "An age control transport protocol for delivering fresh updates in the internet-of-things," in *IEEE International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2019, pp. 1–7.

[22] M. A. Abd-Elmagid and H. S. Dhillon, "Average Peak Age-of-Information Minimization in UAV-Assisted IoT Networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 2003–2008, 2019.

[23] R. Han, Y. Wen, L. Bai, J. Liu, and J. Choi, "Age of information aware uav deployment for intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2705–2715, 2022.

[24] I. Sorkhoh, C. Assi, D. Ebrahimi, and S. Sharafeddine, "Optimizing information freshness for mec-enabled cooperative autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2021.

[25] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis, "Age and value of information: Non-linear age case," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 326–330.

[26] Y. Sun and B. Cyr, "Sampling for data freshness optimization: Non-linear age functions," *Journal of Communications and Networks*, vol. 21, no. 3, pp. 204–219, 2019.

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

[28] 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*, vol. 28, no. 5, pp. 2215–2228, 2020.

[29] Y. Sun, Y. Polyanskiy, and E. Uysal, "Sampling of the wiener process for remote estimation over a channel with random delay," *IEEE Transactions on Information Theory*, vol. 66, no. 2, pp. 1118–1135, 2020.

[30] M. Klügel, M. H. Mamduhi, S. Hirche, and W. Kellerer, "Aoi-penalty minimization for networked control systems with packet loss," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 189–196.

[31] J. P. Champati, M. H. Mamduhi, K. H. Johansson, and J. Gross, "Performance characterization using aoi in a single-loop networked control system," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 197–203.

[32] O. Ayan, M. Vilgelm, M. Klügel, S. Hirche, and W. Kellerer, "Age-of-information vs. value-of-information scheduling for cellular networked control systems," in *ACM/IEEE International Conference on Cyber-Physical Systems*, 2019.

[33] O. Ayan, H. M. Gürsu, S. Hirche, and W. Kellerer, "Aoi-based finite horizon scheduling for heterogeneous networked control systems," in *IEEE Global Communications Conference (GLOBECOM)*, 2020, pp. 1–7.

[34] D. P. Bertsekas, *Dynamic programming and optimal control*, 4th ed. Athena Scientific, 1995, vol. 1.

[35] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of monte carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, pp. 1–43, 2012.

[36] V. Tripathi and E. Modiano, "Age debt: A general framework for minimizing age of information," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2021, pp. 1–6.

[37] E. N. Gilbert, "Capacity of a burst-noise channel," *The Bell System Technical Journal*, vol. 39, no. 5, pp. 1253–1265, 1960.

[38] 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, pp. 1–15.

[39] A. Bildea, O. Alphand, F. Rousseau, and A. Duda, "Link quality estimation with the gilbert-elliot model for wireless sensor networks," in *IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2015, pp. 2049–2054.

[40] G. Sharma, A. Hassan, and A. Dholakia, "Performance evaluation of burst-error-correcting codes on a gilbert-elliott channel," *IEEE Transactions on Communications*, vol. 46, no. 7, pp. 846–849, 1998.

[41] H. S. Wang and P.-C. Chang, "On verifying the first-order markovian assumption for a rayleigh fading channel model," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 2, pp. 353–357, 1996.

[42] H. S. Wang and N. Moayeri, "Finite-state markov channel-a useful model for radio communication channels," *IEEE Transactions on Vehicular Technology*, vol. 44, no. 1, pp. 163–171, 1995.

[43] K. J. Åström and R. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*, 2008.

[44] N. Abramson, "The aloha system: Another alternative for computer communications," ser. AFIPS '70 (Fall). Association for Computing Machinery, 1970, p. 281–285. [Online]. Available: https://doi.org/10.1145/1478462.1478502

[45] L. G. Roberts, "Aloha packet system with and without slots and capture," *ACM SIGCOMM Computer Communication Review*, vol. 5, no. 2, pp. 28–42, 1975.

[46] H. Chen, Y. Gu, and S.-C. Liew, "Age-of-information dependent random access for massive iot networks," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2020, pp. 930–935.

[47] O. Ayan, P. Kutsevol, H. Y. Özkan, and W. Kellerer, "Semantics- and task-oriented scheduling for networked control systems in practice," *IEEE Access*, vol. 10, pp. 115 673–115 690, 2022.

[48] F. Capozzi, G. Piro, L. Grieco, G. Boggia, and P. Camarda, "Downlink packet scheduling in lte cellular networks: Key design issues and a survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 678–700, 2013.

[49] L. Zou, Z. Wang, Q.-L. Han, and D. Zhou, "Moving horizon estimation for networked time-delay systems under round-robin protocol," *IEEE Transactions on Automatic Control*, vol. 64, no. 12, pp. 5191–5198, 2019.

[50] V. Tripathi, R. Talak, and E. Modiano, "Information freshness in multihop wireless networks," *IEEE/ACM Transactions on Networking*, pp. 1–16, 2022.

[51] S. M. Ross, *Introduction to probability models (11th ed.)*. Academic press, 2014.

[52] O. Ayan, M. Vilgelm, and W. Kellerer, "Optimal scheduling for discounted age penalty minimization in multi-loop networked control," in *IEEE Consumer Communications & Networking Conference*, 2020.