Comprehensive Optimal Network Scheduling Strategies for Wireless Control Systems

RUIJIE FU and LANCONG GUO, the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China
AN ZOU, the University of Michigan – Shanghai Jiao Tong University Joint Institute, Shanghai Jiao Tong University, China
CAILIAN CHEN and XINPING GUAN, the Department of Automation and the Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai Engineering Research Center of Intelligent Control and Management, Shanghai Jiao Tong University, China
YEHAN MA*, the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China

Although wireless control is one of the key technologies for future industries, most wireless networks are only used for monitoring. When wireless networks are applied to transmit control commands, the uncertain link qualities and limited network resources may destroy the performance of multi-loop control systems. Hence, it is critical to allocate these resources to optimize the control performance as the network condition changes and plants evolve. This paper presents comprehensive optimal scheduling strategies for wireless control systems based on adaptive dynamic programming. Firstly, we propose an effective adaptive dynamic programming scheduling (ADPS) strategy to solve the optimal scheduling problem based on the single-step control performance at run-time while significantly reducing computational complexity. Moreover, to overcome the "short-sightedness" of single-step performance prediction, we extend ADPS to ADPS-m (**m**ulti-step prediction), which optimizes multi-step performance by incorporating a longer-horizon evolution of the plants. Furthermore, we propose ADPS-H (**H**eterogeneous flow scheduling) to support heterogeneous flows with different data rates and sizes, and ADPS-H-m (**m**ulti-step prediction for **H**eterogeneous flow scheduling), which schedules heterogeneous flows in a longer prediction horizon. We prove that all these scheduling strategies can achieve optimality and stability under mild assumptions. Extensive experiments integrating TOSSIM and MATLAB/Simulink are performed to evaluate all of the proposed methods in case studies of four- and ten-loop control systems. The simulation results demonstrate that these strategies can effectively improve the control performance at lower computing costs under both cyber and physical disturbances. Under the noise level of -76 dBm, for the four-loop case, ADPS achieves the same control performance as the linear programming while saving 99.5% of the execution time. ADPS-m further improves the control performance by up to 27.0% compared with ADPS at the prediction horizon of 3, and ADPS-H-m improves the performance by up to 32.3% and 8.4% compared with round-robin and ADPS-H, respectively. And the ten-loop case indicates the effectiveness and scalability of the proposed approaches.

*Corresponding author

Authors' addresses: Ruijie Fu, Lancong Guo, Yehan Ma, {furj2022, sjtuglc, yehanma}@sjtu.edu.cn, the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University; An Zou, an.zou@sjtu.edu.cn, the University of Michigan – Shanghai Jiao Tong University Joint Institute, Shanghai Jiao Tong University; Cailian Chen, Xinping Guan, {cailianchen, xpguan}@sjtu.edu.cn, the Department of Automation and the Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai Engineering Research Center of Intelligent Control and Management, Shanghai Jiao Tong University, 800 Dongchuan RD. Minhang District, Shanghai, China.

2378-962X/2024/1-ART1 $15.00
https://doi.org/10.1145/3685933

CCS Concepts: • **Networks → Packet scheduling**; • **Computer systems organization → Sensors and actuators**; **Sensor networks**.

Additional Key Words and Phrases: Wireless network scheduling, cyber-physical systems, optimal scheduling, control performance, wireless control systems, adaptive dynamic programming

ACM Reference Format:
Ruijie Fu, Lancong Guo, An Zou, Cailian Chen, Xinping Guan, and Yehan Ma. 2024. Comprehensive Optimal Network Scheduling Strategies for Wireless Control Systems. *ACM Trans. Cyber-Phys. Syst.* 1, 1, Article 1 (January 2024), 31 pages. https://doi.org/10.1145/3685933

1 INTRODUCTION

In the past decades, industrial automation has been powered by leading technologies, such as the Internet of Things (IoT) and wireless control. To lower deployment and maintenance costs, computation platforms may communicate with the sensors/actuators of many physical plants in the field through wireless networks [28]. However, communication in IoT-based industrial control suffers from limited network resources and harsh operation environment [2]. Moreover, since industrial automation operates multiple industrial plants at scale, it is imperative to optimally allocate network resources to different control loops at run-time. Firstly, as the references and time evolve, the physical plant states vary in patterns, and the network transmission has various impacts on controlling the physical plants. For example, when a physical plant is in its steady state, the physical states and control inputs are around their equilibrium point. In this case, the performance gain of reliable transmission at high network resource cost is insignificant. In contrast, when a physical plant is in its transient process, the control performance might differ depending on wireless conditions. Secondly, the natural dynamics and kinetics of the physical plants in different control loops might vary. For example, the communication latency may significantly impact control performance for a control loop with short time constants. Therefore, it is necessary to allocate network resources to optimize the control performance at run-time for IoT-based control, which faces the following challenges.

To capture and respond to the varying cyber and physical conditions at run-time, the online decision in the scheduler for network resource allocations must be lightweight. Meanwhile, in most current research, the timescales of network resource allocation and the control performance prediction are identical, overlooking the differences between the discrete/logistic nature of the network and the continuous/kinetics nature of the physical plant control. For example, offline network allocations always consider the control performance metrics over an infinite prediction horizon. Similarly, the network allocation over a coordination period is based on the control performance prediction over the next coordination period, which sometimes is selected to be equal to the control period. Moreover, with the in-depth advancement of Industry 4.0, massive heterogeneous sensors and actuators, which can create heterogeneous data flows with different control periods and data sizes, are introduced in the industrial field [35]. The versatility of devices and communications aggravates the scheduling complexity, such as allocating network resources to the control loops with different control periods when other control loops require various network resources. Therefore, heterogeneous data flows should also be well incorporated into the network scheduling of IoT-based control systems.

Targeting the challenges in industrial automation, this paper proposes a series of network scheduling strategies to optimize the overall control performance at run-time for IoT-based control systems. The key contributions are listed as follows:

- We formulate the network scheduling problem as an integer programming problem. To avoid the prohibitively high complexity of solving such an integer programming scheduling problem, we propose an adaptive dynamic programming scheduling (ADPS) approach, which can optimally schedule the network under cyber-physical comprehensive measurements with reduced computational complexity to $O(2N(L+1))$; where L is the limited network resource represented by the number of time slots, N the number of control loops.

- Instead of applying identical time scales for resource allocation and performance prediction, we extend the proposed ADPS to ADPS-m (**m**ulti-step prediction), which incorporates network resource allocation with the multi-step control performance prediction, to further improve the control performance.
- For heterogeneous data flows with different control rates and data sizes, the ADPS algorithm is improved to ADPS-H (**H**eterogeneous flow scheduling) and ADPS-H-m (**m**ulti-step prediction for **H**eterogeneous flow scheduling), which support more sophisticated scenarios and versatile system variations.
- We provide stability analyses for these comprehensive scheduling strategies and show that the optimal scheduling strategies can guarantee the mean-square stability of the multi-loop control system under mild assumptions.
- Extensive experiments are performed on the wireless cyber-physical system simulator, which integrates TOSSIM and MATLAB/Simulink to demonstrate the feasibility and superiority of the proposed approaches.

The rest of this paper is organized as follows. Sec. 2 reviews the related work. Sec. 3 presents the modeling of the IoT-based control system and network scheduling. In Sec. 4, we propose the ADPS approach to solve the scheduling problem more concisely and efficiently. In Sec. 5, we extend ADPS to ADPS-m for multi-horizon performance optimization. In Sec. 6, ADPS-H and ADPS-H-m are introduced for heterogeneous data flows. The stability analysis of ADPS-H-m is provided in Sec. 7. The evaluation is presented in Sec. 8. Finally, Sec. 9 concludes the paper.

2 RELATED WORK

IoT-based control over wireless networks in the context of industrial automation faces serious challenges due to the nondeterminism and limited resources of the network, which have facilitated a variety of research in both network and control communities to improve the system performance.

Aiming at making better utilization of network resources under various network conditions in industrial applications, several network configuration solutions have been proposed. Blink protocol provides hard guarantees on end-to-end deadlines of received packets in multi-hop low-power wireless networks, while seamlessly handling changes in application requirements and network state [47]. SchedEx, a generic heuristic scheduling algorithm extension, guarantees a user-defined end-to-end reliability [12]. REACT reduces the latency and energy cost of network reconfiguration, improving the agility of wireless networks in dynamic environments [16]. Shi et al. [39] presented distributed graph routing and autonomous scheduling solutions that allow the devices to compute their own graph routes and schedules. However, the above studies only schedule the network from a networking perspective, while the influence of physical plant control is ignored.

Considering the control performance of physical plants, network scheduling strategies by network-control co-design have been studied. In terms of the offline and online period assignment, Saifullah et al. proposed offline [38] and online [34] optimal sampling period selection to optimize the control performance. Kim et al. [22] investigated heterogeneous sampling period assignment when multiple loops are controlled through a WiFi network. However, none of the works [22, 34, 38] consider the effects of link quality on the control performance. Eisen et al. established an optimal network scheduling with respect to a control operation cost in 5G [14] and WiFi [13] networks. Ma et al. [29] proposed an optimal dynamic scheduling strategy to optimize the performance of multi-loop control systems based on predictions of both link quality and control performance over the IEEE 802.15.4 network. Antunes et al. [4, 5] proposed dynamic programming rollout strategies to select which node should access the network at each transmission based on the states of plants and sensors.

Among the above state-of-the-art, static scheduling approaches [22, 38] for different network standards are proposed to optimize the static control performance metrics over an infinite horizon, which cannot adapt to varying and harsh cyber and physical conditions during run-time. One-step performance optimization [13, 14, 29] may overlook the potential performance improvement by considering the dynamic and kinetics of the industrial plants for longer horizons. Antunes et al. [4, 5] considered finite and infinite performance prediction in an ideal network that only one node can access without failure in each control period. Besides, most of the above work formulates a computationally expensive optimization problem, followed by providing solutions based on relaxation and approximation. Furthermore, most work assumes the identical size of data flows. In this work, we first focus on providing lightweight scheduling solutions in IEEE 802.15.4 network without loss of optimality, and we extend the ADPS approach to explore a more comprehensive view of multi-step prediction and heterogeneous data flows.

3 MODELING OF IOT-BASED CONTROL SYSTEM AND NETWORK SCHEDULING

In this section, we present the modeling of IoT-based control systems. In an IoT-based control system, N loops share a wireless network with limited resources. Our goal is to allocate the network resources at run-time to optimize the control performance based on the states of plants and the network.

3.1 Multi-loop Control System

Control loop $i \in \{1, 2, ..., N\}$, is modeled as a discrete-time linear time-invariant (DT-LTI) system, the state space equation of which can be expressed as:

$$x_i(t+1) = A_i x_i(t) + B_i u_i(t), u_i(t) = K_i \hat{x}_i(t), \tag{1}$$

where $x_i(t)$ and $u_i(t)$ are the state vector and control command vector of loop i at time index t, A_i and B_i are parameter matrices, and K_i is the state feedback gain.

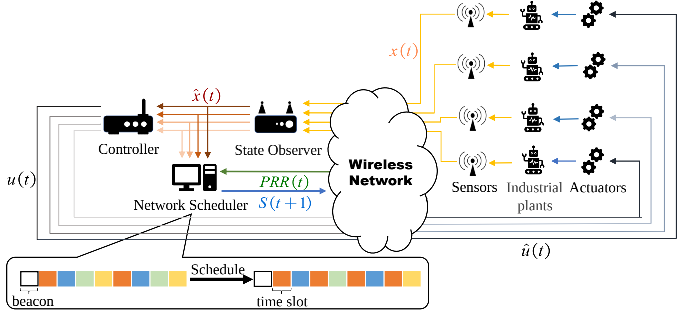


Fig. 1. Network scheduling of the IoT-based control system

The sensors and actuators of industrial plants communicate with the controllers via the shared wireless network. As shown in Fig. 1, after obtaining the plant states and link quality information, the controller generates control command $u_i(t)$ according to the state estimation $\hat{x}_i(t)$ from the state observer in each control period T_i. The network scheduler, which generates schedules for the wireless network, is co-located with the controller [40], such that it can capture the physical plant information, including physical models A_i, B_i, the control law K_i, and the run-time physical states $\hat{x}_i(t)$ [13], without extra implementation costs. Depending on whether the downstream transmission of the control commands is successful (i.e., reception or loss), we obtain the following system dynamics for $x_i(t+1)$ as

$$x_i(t+1) = \begin{cases} x_i^c(t+1) = A_i x_i(t) + B_i \hat{u}_i^c(t) & \text{reception} \\ x_i^o(t+1) = A_i x_i(t) + B_i \hat{u}_i^o(t) & \text{loss,} \end{cases} \tag{2}$$

where

$$\hat{u}_i^c(t) = u_i(t),$$
$$\hat{u}_i^o(t) = \hat{u}_i(t-1).$$

The control commands $\hat{u}_i(t-1)$ will be applied by the actuator in the event of a network failure at time t. In this work, we restrict our attention to the downstream network scheduling for control commands [27, 36, 37] while the upstream sensing is assumed to occur over an ideal channel [27, 37]. We keep this simpler model for mathematical coherence in our problem formulation and stability proof in Secs. 4–7. Please note that we lift this assumption and set the uplink to be lossy in Sec. 8. The upstream scheduling is another important issue for practical wireless control systems. Interested readers can refer to [13, 14] and references therein for details about upstream sensing packet scheduling. We will extend the optimal scheduling to both downstream and upstream in our future work.

We assume the applied input $\hat{u}_i(t-1)$ of the physical plant is assumed to be known by the controller in order to compute the open-loop cost [27, 32, 33]. As the upstream is assumed to occur over an ideal channel, we may regard the binary of whether the control data received by the actuator as an additional state of the physical plant is transmitted over the ideal upstream channel. In the future, we will extend optimal scheduling to scenarios that the controller does not need the knowledge of $\hat{u}_i(t-1)$.

The control performance $J(x_i(t))$ of loop i is represented by the quadratic control cost J as [3]

$$J\big(x_i(t)\big) = x_i^T(t) W_i x_i(t), \tag{3}$$

where W_i is a positive definite matrix.

3.2 Wireless Network

In the IoT-based control system, the sensors, actuators, and controllers are communicated via the wireless sensor-actuator network. The IEEE 802.15.4 standard (the basis of Zigbee, ISA100.11a, WirelessHART, and MiWi) is widely applied in industrial applications due to its low power consumption and high reliability. We assume a WSAN with star topology comprising a coordinator and devices equipped with IEEE 802.15.4 radio, where each device can directly communicate with the coordinator [14, 18, 19]. The downlink is lossy, while the upstream is assumed to occur over an ideal channel in order to investigate the impacts of downstream control command transmissions [36] in wireless control systems. In IEEE 802.15.4 MAC, a superframe consists of beacons, contention-access period (CAP), and contention-free period (CFP), the length of which could be free design parameters in the modified version of the standard [6]. Since we are targeting the control systems with real-time requirements, we focus on scheduling the guaranteed time slots (GTSs) in CFP. The wireless nodes synchronize by receiving and decoding the beacon frames. In IEEE 802.15.4 slotted mode, the synchronization can be maintained by slotted communication among wireless nodes according to the standard [1]. In the resultant time division multiple access (TDMA) network, the superframe repeats itself to transfer data flows periodically. As the TDMA defines deterministic access in the data link layer, the deadline misses caused by the packet delay can be regarded as packet losses modeled in Sec. 3.1.

The link quality of the wireless network can be indicated by the packet reception ratio (PRR). We predict the link quality at the granularity of a superframe as the scheduling is done on a superframe basis. As a superframe is short, typically tens to hundreds of milliseconds, and we apply Holt's additive trend prediction algorithm [17, 42], which essentially filters PRRs over time series, the predicted link quality is assumed to remain constant within the same superframe. Holt's additive trend prediction method is employed to predict the PRR of the next ℓ superframes as follows.

$$\begin{aligned} G_i(t) =& \xi PRR_i(t) + (1-\xi)\left(G_i(t-1) + F_i(t-1)\right) \\ F_i(t) =& \psi\left(G_i(t) - G_i(t-1)\right) + (1-\psi)F_i(t-1) \\ \widehat{PRR}_i(t+\ell|t) =& G_i(t) + \ell F_i(t), \end{aligned}$$

where $PRR_i(t)$ is the current measured PRR of link i, $G_i(t)$ denotes an estimation of the current level of the series, $F_i(t)$ represents an estimation of the current trend, ℓ is a positive integer representing the superframes ahead, $\widehat{PRR}(t+\ell|t)$ is the predicted PRR ℓ superframes ahead, and ξ and ψ $(0 < \xi, \psi < 1)$ are the level and slope smoothing coefficient, respectively. After the link prediction, the estimated failure ratio $\beta_i(t)$ of the link of loop i at time t can be expressed as

$$\beta_i(t) = 1 - \widehat{PRR}_i(t). \tag{4}$$

Assumption 3.1. *We assume that the packet loss of each link is independent and identically distributed (i.i.d.) in order to prove the stability of WNCS. The i.i.d. assumption can significantly simplify control design and stability analysis [7, 30, 36]. Please note that we lift the i.i.d packet loss assumption in the evaluation by simulating stochastic packet loss according to a realistic wireless channel model in TOSSIM seeded by signal traces from real-world wireless networks.*

3.3 Network Scheduling in IoT-based Control

The network scheduler allocates GTSs to transfer the data flows of each control loop based on control cost and network quality in each coordination period T_c. Most existing works [7, 19, 29] optimizing single-step or infinite-horizon control performance assume that all loops in the multi-loop system have the same control period as well as the control data size. However, plants have multiple sampling and control periods in real industrial scenarios due to their different physical properties and various types of sensors and actuators, resulting in heterogeneous data flows [20, 46]. Besides, network scheduling approaches only based on single-step control performance optimization are short-sighted. In contrast, a multi-step performance prediction can provide a better scheduling criterion. Therefore, we focus on a more general objective function for the multi-step control cost of the heterogeneous data flows. At time t, the expectation of the overall control cost $\mathcal{J}(\cdot)$ can be defined as

$$\mathbb{E}\Big(\mathcal{J}\big(x(t), \beta(t), \mathcal{T}, m, \mathcal{H}\big)\Big) = \sum_{i=1}^{N} \mathbb{E}\Big(\mathcal{J}_i\big(x_i(t), \beta_i(t), T_i, m_i, H_i\big)\Big) \tag{5}$$

where $\mathbb{E}$ is the expectation operator, $\mathcal{J}_i(\cdot)$ is the control cost of loop i which has specific forms in different scenarios and will be introduced in the following sections. The network condition is represented by $\beta(t) = \{\beta_1(t), \beta_2(t), ..., \beta_N(t)\}$, which contains the link failure ratios of N loops. $\mathcal{T} = \{T_1, T_2, ..., T_N\}$ and $\mathcal{H} = \{H_1, H_2, ..., H_N\}$ are the sets of the control periods T_i and the control data sizes H_i for each loop i, respectively. The data size H_i is expressed as the number of time slots required for a single transmission. The prediction step m of the overall control performance results in different prediction step of loop i denoted as m_i, which will be detailed in Sec. 6.

For the objective function $\mathbb{E}\big(\mathcal{J}(\cdot)\big)$, we minimize it by dynamically scheduling the number of transmissions $n_i(t+1)$ for the next control period of each loop and the optimal scheduling problem can be expressed as:

$$\begin{aligned} \min_{n_i(t+1)} \quad & \mathbb{E}\Big(\mathcal{J}\big(x(t), \beta(t), \mathcal{T}, m, \mathcal{H}\big)\Big) \\ s.t. \quad & \sum_{i=1}^{N} H_i n_i(t+1) \leq L \\ & n_i(t+1) \in \{0, 1, ..., L\} \\ & \forall i \in \{1, 2, ..., N\} \end{aligned} \tag{6}$$

where $n_i(t+1)$ is the number of transmissions of loop i for its next control period T_i, L is the limited network resource represented by the number of slots in a superframe. We assume that the number of time slots L in one superframe is larger than or equal to the maximum H_i in $\mathcal{H}$ so that the control data with any size can be allowed to transmit within one superframe. According to Köppe et al. [23], the network scheduling problem (6) is a NP-hard problem of resource-constrained scheduling. In the following sections, we first present the ADPS which solves problem (6) based on the single-step prediction result for loops with the same control period and data size, i.e., $T_i = T_j, m_i = m_j = 1$, and $H_i = H_j, \forall i, j \in \{1, 2, ..., N\}$ optimally with significantly reduced computational complexity in Sec. 4. Furthermore, we extend ADPS to the ADPS-m with the multi-step prediction of the control cost in Sec. 5, i.e., $T_i = T_j, m_i > 1$, and $H_i = H_j, \forall i, j \in \{1, 2, ..., N\}$. In Sec. 6, we present the ADPS-H ($T_i \neq T_j, m_i = m_j = 1$, and $H_i \neq H_j, \exists i, j \in \{1, 2, ..., N\}$) and ADPS-H-m ($T_i \neq T_j, m_i > 1$, and $H_i \neq H_j, \exists i, j \in \{1, 2, ..., N\}$) for optimally scheduling heterogeneous data flows and show that ADPS, ADPS-m, and ADPS-H are all special cases of the ADPS-H-m.

4 ADAPTIVE DYNAMIC PROGRAMMING SCHEDULING

The state-of-the-art cyber-physical system modeling and co-design have explored the interdependencies between physical and communication systems. Actually, we find that the physical and communication systems are interdependent not only in modeling but also in the process of solving scheduling problems, which is the highlight of our ADPS algorithm. While all the loops have the same control period T_i, we set the coordination period $T_c = T_i$. At time $t-1$, as $T_i = T_j, m_i = m_j = 1$, and $H_i = H_j = 1, \forall\, i, j \in \{1, 2, ..., N\}$, the symbols $\mathcal{T}$, m, and $\mathcal{H}$ in $\mathbb{E}\big(\mathcal{J}(\cdot)\big)$ are omitted for simplification, and the expectation of overall control cost $\mathbb{E}\big(\mathcal{J}\big(x(t-1), \beta(t-1)\big)\big)$ can be expressed as Eq. (7). Please note that we set the base time as $t-1$ instead of t and then predict the control cost for time t in this section to simplify the equations by avoiding too many '$t+1$'.

$$\mathbb{E}\Big(J\big(x(t)\big)\Big) = \sum_{i=1}^{N} \Big(J\big(x_i^c(t)\big)\big(1 - \beta_i^{n_i(t)}(t)\big) + J\big(x_i^o(t)\big)\beta_i^{n_i(t)}(t)\Big), \tag{7}$$

After separating the control component $\Lambda(t)$ from the communication component $\Theta(t)$, then the one-step prediction of the control cost (7) can be rewritten as Eq. (8).

$$\mathbb{E}\Big(J\big(x(t)\big)\Big) = \Lambda(t)\Theta^T(t) = \sum_{i=1}^{N} \big(\Lambda_{0,i}(t) + \Lambda_{1,i}(t)\beta_i^{n_i(t)}(t)\big), \tag{8}$$

where

$$\begin{cases} \Lambda(t) = \big[[\Lambda_{0,1}(t)\ \Lambda_{1,1}(t)], ..., [\Lambda_{0,N}(t)\ \Lambda_{1,N}(t)]\big] \\ \Theta(t) = \big[[1\ \beta_1^{n_1(t)}(t)], ..., [1\ \beta_N^{n_N(t)}(t)]\big] \\ \Lambda_{0,i}(t) = J\big(x_i^c(t)\big) \\ \Lambda_{1,i}(t) = J\big(x_i^o(t)\big) - J\big(x_i^c(t)\big). \end{cases}$$

Considering the allocation of each time slot as a discrete process, the benefit of the control cost for loop i at j-th slot is calculated as

$$\Delta\mathbb{E}\Big(J\big(x_i(t)\big)\Big) = \sum\big(\Phi_i(j)\odot\theta_i - \Phi_i(j)\big) = \Lambda_{1,i}(t)\beta_i^{n_i(t,j)}(t)(\beta_i(t)-1), \tag{9}$$

where

$$\begin{cases}\Phi_i(j) & = [\Lambda_{0,i}(t) \quad \Lambda_{1,i}(t)\beta_i^{n_i(t,j)}(t)] \\ \theta_i & = [1 \quad \beta_i(t)].\end{cases}$$

$\Phi_i(j)$ consists of the predicted control components $\Lambda_{0,i}(t)$ and $\Lambda_{1,i}(t)$ in Eq. (9), $\beta_i(t)$, and $n_i(t,j)$. $n_i(t,j)$ is the number of time slots that has been allocated to loop i **before** the slot j's allocation. θ_i means the loop i's communication component of unit time slot as we assume $\beta_i(t)$ keeps unchanged during this superframe duration. $\Phi_i(j)\odot\theta_i$ is the Hadamard product of $\Phi_i(j)$ and θ_i. $\sum$ indicates the sum of matrix elements.

We complete the iteration of calculating control cost benefit $\Delta\mathbb{E}\big(J\big(x_i(t)\big)\big)$ during the allocation of each time slot. Demirel et. al [10] have proven that for a fixed control period and delay, the optimal control loss is monotone decreasing in the deadline-constrained reliability, which indicates $\Lambda_{1,i} \geq 0, \forall i \in \{1,2,...,N\}$. As $0 \leq \beta_i(t) \leq 1$, the value of $\Delta\mathbb{E}\big(J\big(x_i(t)\big)\big)$ in Eq. (9) is nonpositive. Therefore, the minimum sum of control benefit $\Delta\mathbb{E}\big(J\big(x(t)\big),j\big)$ after the allocation of j-th slot is

$$\Delta\mathbb{E}\big(J\big(x(t)\big),j\big) = \min_i\Big(\Delta\mathbb{E}\big(J\big(x(t)\big),j-1\big) + \Delta\mathbb{E}\big(J\big(x_i(t)\big)\big)\Big), \tag{10}$$

The optimal gradient for the j-th time slot is the loop whose control benefit shown in Eq. (9) results in the minimum $\Delta\mathbb{E}\big(J\big(x(t)\big),j\big)$. By denoting this optimal gradient as i_j^*, we can update $\Phi_i(j)$ and $n_i(t,j)$ with the following procedures,

$$\begin{cases}\Phi_i(j) \longleftarrow [\Lambda_{0,i}(t) \quad \Lambda_{1,i}(t)] & j=1 \\ n_i(t,j) \longleftarrow 0 & j=1 \\ S(t) \longleftarrow \varnothing & j=1 \\ \Phi_{i_j^*}(j+1) \longleftarrow \Phi_{i_j^*}(j)\odot\theta_i & j>1, i=i_j^* \\ \Phi_i(j+1) \longleftarrow \Phi_i(j) & j>1, i\neq i_j^* \\ n_{i_j^*}(t,j+1) \longleftarrow n_{i_j^*}(t,j)+1 & j>1, i=i_j^* \\ n_i(t,j+1) \longleftarrow n_i(t,j) & j>1, i\neq i_j^* \\ S(t) \longleftarrow [S(t) \quad i_j^*] & j>0.\end{cases} \tag{11}$$

When all time slots have been allocated, the final $S(t)$ is the optimal schedule for the next coordination period. The "⟵" indicates the schedule updating in (11).

As $\Lambda_{0,i}(t)$ corresponds to $\beta_i^0 = 1$, it will be ignored in the actual programming. The benefit of Eqs. (8) and (11) is the guaranteed independence of physical and network system metrics, ensuring dynamic programming optimality. Finally, we can get the optimal scheduling through the statistics of i_j^*. Fig. 2 shows the diagram of the ADPS algorithm at time t with the number of control loops $N = 6$ and the number of time slots $L = 5$. The blue block in the j-th row with $\Phi_i(j)$ indicates the transmission schedule assigned to the j-th slot, and the red arrow "↓" between j-th row and $(j+1)$-th row indicates the update process of $\Phi_i(j+1)$ after loop i is assigned the j-th slot. The dark green block in the $(j+1)$-th row which is pointed to by the red "↓" indicates the updated $\Phi_i(j+1)$ for the optimal gradient i_{j+1}^* calculation. We magnify the initialization of $\Phi_i(1)$ and the update of $\Phi_i(5)$ after $i_4^* = 6$. The update progress of the scheduling has been given on the left as we mark i_j^* with blue (before the update) and dark green (after the update). The final scheduling result $S(t+1) = [2\ 4\ 3\ 6\ 2]$.

In addition to calculating $x^c(t+1)$ and $x^o(t+1)$ with the complexity of $O(2N)$, ADPS needs to complete gradient calculations $(O(NL))$, and searching the maximum $\Delta\mathbb{E}\big(J\big(x_i(t)\big)\big)$ with the

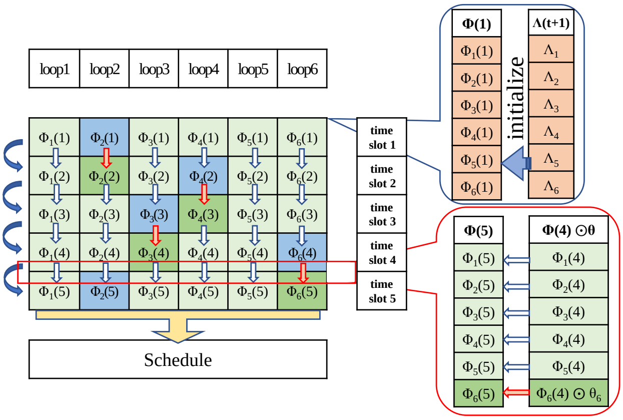

loop1 loop2 loop3 loop4 loop5 loop6
Φ1(1) Φ2(1) Φ3(1) Φ4(1) Φ5(1) Φ6(1)
Φ1(2) Φ2(2) Φ3(2) Φ4(2) Φ5(2) Φ6(2)
Φ1(3) Φ2(3) Φ3(3) Φ4(3) Φ5(3) Φ6(3)
Φ1(4) Φ2(4) Φ3(4) Φ4(4) Φ5(4) Φ6(4)
Φ1(5) Φ2(5) Φ3(5) Φ4(5) Φ5(5) Φ6(5)
Schedule
time slot 1
time slot 2
time slot 3
time slot 4
time slot 5
Φ(1)
Φ1(1) Φ2(1) Φ3(1) Φ4(1) Φ5(1) Φ6(1)
initialize
Λ(t+1)
Λ1 Λ2 Λ3 Λ4 Λ5 Λ6
Φ(5)
Φ1(5) Φ2(5) Φ3(5) Φ4(5) Φ5(5) Φ6(5)
Φ(4) ⊙θ
Φ1(4) Φ2(4) Φ3(4) Φ4(4) Φ5(4) Φ6(4) ⊙ θ6

Fig. 2. The diagram of ADPS algorithm at time t.

Algorithm 1: ADPS Algorithm

```
Input: $\theta_i$, $L$, $N$, $x(t)$, $i \in \{1, 2, ..., N\}$.
Output: $S(t+1)$.
calculate $x^c(t+1)$, $x^o(t+1)$, and $\Lambda(t+1)$ in Eq. (8);
$S(t+1) = \varnothing$, $\Delta\mathbb{E}\big(J\big(x(t+1), 1\big)\big) = +\infty$;
$\Phi(1) \leftarrow \Lambda(t+1)$;
for all $j \in \{1, 2, ...L\}$ do
    for all $i \in \{1, 2, ..., N\}$ do
        $\Delta\mathbb{E}\big(J\big(x_i(t+1)\big)\big) = \sum \big(\Phi_i(j) \odot \theta_i - \Phi_i(j)\big)$;
    end
    for all $i \in \{1, 2, ..., N\}$ do
        $\Delta\mathbb{E}\big(J\big(x(t+1)\big), j\big) \leftarrow$
          $\min_i \Big(\Delta\mathbb{E}\big(J\big(x(t+1)\big), j\big), \Delta\mathbb{E}\big(J\big(x(t+1)\big), j-1\big) + \Delta\mathbb{E}\big(J_i\big(x_i(t+1)\big)\big)\Big)$;
    end
    $i_j^* \leftarrow i$ with the minimum $\Delta\mathbb{E}\big(J\big(x(t+1)\big), j\big)$;
    update $\Phi_i$: $\Phi(j+1) \leftarrow \begin{bmatrix} 1 & \dots & i_j^* & \dots & N \\ \Phi_1(j) & \dots & \Phi_{i_j^*}(j) \odot \theta_{i_j^*} & \dots & \Phi_N(j) \end{bmatrix}$;
    $S(t+1) \leftarrow [S(t+1) \quad i_j^*]$.
end
return $S(t+1)$.
```

complexity of $O(NL)$ in allocating L slots. Thus the computational complexity of ADPS is $O(2N(L+1))$. In contrast, the LP relaxation given in the study of Ma et, al. [29] is $O\big((N(L+1))^3 \ln(N(L+1))\big)$ and the brute force solution is $O(N^L)$. The proposed ADPS algorithm benefits significantly reduced computational complexity.

5 MULTI-STEP PERFORMANCE PREDICTION: ADPS-M

The ADPS proposed in Sec. 4 optimally allocates the network resources based on the one-step performance prediction. This scheduling strategy is "short-sighted" because it derives scheduling results based on single-step performance predictions only. To overcome this shortsightedness and get more accurate performance predictions, instead of applying identical timescales for network resource allocation and performance prediction, we extend ADPS to ADPS-m, which incorporates single-step network resource allocation according to the multi-step control performance prediction. In this section, we assume that each control loop has the same control period and prediction step, i.e., $T_i = T_j = T_c$, $m_i = m_j = m > 1$, $\forall\, i, j \in \{1, 2, ..., N\}$ where m is called the number of prediction step of the overall N loops. The heterogeneous case will be discussed in Sec. 6. Fig. 3 presents the relationship between the one-step forward scheduling and multi-step performance prediction intervals in ADPS-m. In the coordination period, which begins at time t (the blue block), ADPS-m establishes the optimal scheduling $S(t+1)$ for the next T_c (the green block) based on future m-step performance predictions (the yellow blocks) by assuming the same schedule is applied during the prediction interval, i.e., $S(t+1) = S(t+2) = ... = S(t+m)$. The above operations repeat for each coordination period T_c.

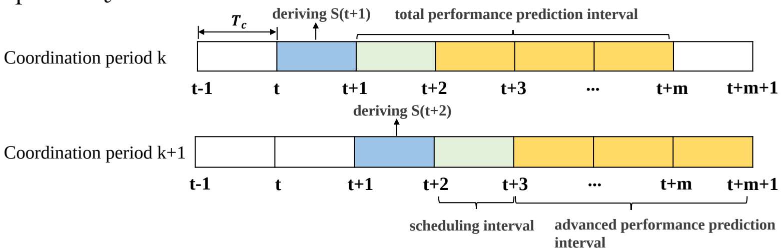


Fig. 3. Diagram of one-step-forward scheduling and multi-step prediction.

For a deterministic $x(t)$, the loss and reception of $u(t)$ would result in two possible $x(t+1)$ (x^c and x^o), thus the number of state predictions in $(r+1)$-th step is twice as the number of states in r-th step, i.e., the r-th step prediction would produce $2r$ predicted states. We denote the l-th state in the r-th step's states sequence of loop i as $x_{l,i}(t+r)$. In the sequence, the state with an odd index is estimated based on the successful reception of u_i, and the state estimation with an even index uses the previous control command. Finally, the predicted states updating can be expressed as

$$x_{l,i}(t+r+1) = \begin{cases} A_i x_{(l+1)/2,i}(t+r) + B_i u_{l,i}(t+r), \text{if } l \bmod 2 = 1 \\ A_i x_{l/2,i}(t+r) + B_i u_{l,i}(t+r). \qquad \text{else,} \end{cases} \tag{12}$$

where

$$u_{l,i}(t+r) = \begin{cases} K_i x_{(l+1)/2,i}(t+r), \text{if } l \bmod 2 = 1 \\ u_{l/2,i}(t+r-1). \quad\ \text{else,} \end{cases}$$

and $l \in \{1, 2, ..., 2r\}$. As the schedule $S(t+1)$ is assumed to be applied during the whole prediction interval, the control message is transmitted $n_i(t+1)$ times at r-th step for loop i. The transfer probability $p_{l,i}(t+r)$ from state $x_i(t)$ to the l-th possible physical state $x_{l,i}(t+r)$ at time $t+r$ can be expressed as Eq. (13).

$$p_{l,i}(t+r+1) = \begin{cases} p_{(l+1)/2,i}(t+r)(1-\beta_i^{n_i(t+1)}(t+r+1)), \text{if } l \bmod 2 = 1 \\ p_{l/2,i}(t+r)\beta_i^{n_i(t+1)}(t+r+1). \qquad\qquad \text{else.} \end{cases} \tag{13}$$

Fig. 4 shows a diagram of 2-step prediction. The blue arrow indicates the successful reception in this step and the orange arrow indicates the state prediction results under packet loss. $p_{l,i}(t+r)$ above the arrow indicates the probability of transferring from $x_i(t)$ to $x_{l,i}(t+r)$, and the control command $u_{l,i}(t+r-1)$ corresponding to $x_{l,i}(t+r)$ is below the arrow. Referring to Eqs. (12) and (13), the expectation of the control cost $\mathbb{E}\big(J\big(x_i(t+r)\big)\big)$ at the r-th control period is

$$\mathbb{E}\Big(J\big(x_i(t+r)\big)\Big) = \sum_{l=1}^{2r} p_{l,i}(t+r) J\big(x_{l,i}(t+r)\big), \tag{14}$$

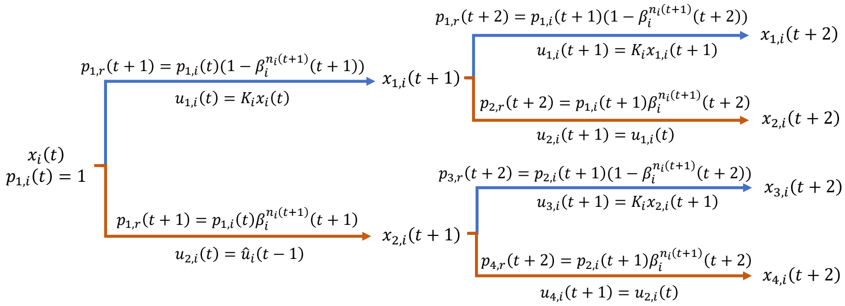

$p_{1,r}(t+2) = p_{1,i}(t+1)(1-\beta_i^{n_i(t+1)}(t+2))$
$u_{1,i}(t+1) = K_i x_{1,i}(t+1)$
$x_{1,i}(t+2)$
$p_{1,r}(t+1) = p_{1,i}(t)(1-\beta_i^{n_i(t+1)}(t+1))$
$u_{1,i}(t) = K_i x_i(t)$
$x_{1,i}(t+1)$
$p_{2,r}(t+2) = p_{1,i}(t+1)\beta_i^{n_i(t+1)}(t+2)$
$u_{2,i}(t+1) = u_{1,i}(t)$
$x_{2,i}(t+2)$
$x_i(t)$
$p_{1,i}(t) = 1$
$p_{3,r}(t+2) = p_{2,i}(t+1)(1-\beta_i^{n_i(t+1)}(t+2))$
$u_{3,i}(t+1) = K_i x_{2,i}(t+1)$
$x_{3,i}(t+2)$
$p_{1,r}(t+1) = p_{1,i}(t)\beta_i^{n_i(t+1)}(t+1)$
$u_{2,i}(t) = \hat{u}_i(t-1)$
$x_{2,i}(t+1)$
$p_{4,r}(t+2) = p_{2,i}(t+1)\beta_i^{n_i(t+1)}(t+2)$
$u_{4,i}(t+1) = u_{2,i}(t)$
$x_{4,i}(t+2)$

Fig. 4. A 2-step state prediction procedure.

where $r \in \{1, 2, ..., m\}$. Similar to ADPS, as $T_i = T_j = T_c$, and $H_i = H_j, \forall\ i, j \in \{1, 2, ..., N\}$, we omit the symbols $\mathcal{T}$ and $\mathcal{H}$ in $\mathbb{E}\big(\mathcal{J}(\cdot)\big)$. Substituting Eq. (14) into Eq. (5), the expected control cost $\mathbb{E}\big(\mathcal{J}_i\big(x_i(t), \beta_i(t), m_i\big)\big)$ of loop i with multi-step prediction can be expressed as Eq. (15).

$$\begin{aligned}\mathbb{E}\Big(\mathcal{J}_i\big(x_i(t), \beta_i(t), m_i\big)\Big) = \mathbb{E}\Big(J_{\boldsymbol{m}}\big(x_i(t)\big)\Big) &= \sum_{r=1}^{m} \alpha^{r-1} \mathbb{E}\Big(J\big(x_i(t+r)\big)\Big) \\ &= \sum_{r=1}^{m} \alpha^{r-1} \sum_{l=1}^{2r} p_{l,i}(t+r) J\big(x_{l,i}(t+r)\big) \\ &= \sum_{r=1}^{m} \sum_{g=0}^{2^r-1} \lambda_{g,r,i} f(\beta_i, g, r)^{n_i(t+1)}, \end{aligned} \tag{15}$$

We define α^{r-1} as the attenuation factor at the r-th control period due to the modeling inaccuracy and the prediction error of link quality, $0 < \alpha \leq 1$. It is worth noting that α is introduced without loss of generality as the expectation of classic finite-horizon state quadratic in [31] is a special case of Eq. (15) when $\alpha = 1$. In order to align with the single-step control performance prediction Eq. (8), we defined $f(\beta_i, g, r) = \prod_{j=0}^{r-1} \beta_i(t+1+j)^{\lfloor \frac{g}{2^j} \rfloor \mod 2}$ as the communication component, and $\lambda_{g,r,i}$ as the control component in the multi-step performance prediction, respectively. For example, with $r = 2, f(\beta_i, 0, 2) = 1, f(\beta_i, 1, 2) = \beta_i(t+1), f(\beta_i, 2, 2) = \beta_i(t+2), f(\beta_i, 3, 2) = \beta_i(t+1) * \beta_i(t+2)$. $\lambda_{g,r,i}$ is the coefficient of $f(\beta_i, g, r)$ in the r-th step prediction, which is related to α as well as $x_{l,i}(t+r)$ and can be derived by combining the like terms of the second line of Eq. (15). The multi-step expected overall control cost can finally be expressed as

$$\mathbb{E}\Big(\mathcal{J}\big(x(t), \beta(t), m\big)\Big) = \Lambda_m(t)\Theta_m^T(t) = \sum_{i=1}^{N} \sum_{r=1}^{m} \sum_{g=0}^{2^r-1} \lambda_{g,r,i} f(\beta_i, g, r)^{n_i(t+1)}, \tag{16}$$

where

$$\begin{cases} \Lambda_m(t) & = \big[\Lambda_{m,1}(t), ..., \Lambda_{m,N}(t)\big] = \\ & \quad \big[[\lambda_{0,1,1}, \lambda_{1,1,1}, ..., \lambda_{0,m,1}, ..., \lambda_{2^m-1,m,1}], ..., [\lambda_{0,1,N}, \lambda_{1,1,N}, ..., \lambda_{0,m,N}, ..., \lambda_{2^m-1,m,N}]\big], \\ \Theta_m(t) & = \big[\Theta_{m,1}(t), ..., \Theta_{m,N}(t)\big] = \\ & \quad \big[[1, f(\beta_1, 1, 1)^{n_1(t+1)}, ..., 1, ..., f(\beta_1, 2^m-1, m)^{n_1(t+1)}], \\ & \quad ..., [1, f(\beta_N, 1, 1)^{n_N(t+1)}, ..., 1, ..., f(\beta_N, 2^m-1, m)^{n_N(t+1)}]\big]. \end{cases}$$

The benefit of the control cost for loop i at j-th slot is

$$\Delta\mathbb{E}\Big(J_{\boldsymbol{m}}\big(x_i(t)\big), j\Big) = \sum \big(\Phi_{m,i}(j) \odot \Theta_i - \Phi_{m,i}(j)\big) = \sum_{r=1}^{m} \sum_{g=0}^{2^r-1} \lambda_{g,r,i} f(\beta_i, g, r)^{n_i(t+1,j)} \big(f(\beta_i, g, r) - 1\big), \tag{17}$$

where

$$\begin{cases}\Phi_{m,i}(j) = [\lambda_{0,1,i}, \lambda_{1,1,i}f(\beta_i, 1, 1)^{n_i(t+1,j)}, ..., \lambda_{0,m,i}, ..., \lambda_{2^m-1,m,i}f(\beta_i, 2^m-1, m)^{n_i(t+1,j)}] \\ \Theta_i = [1, f(\beta_i, 1, 1), ..., 1, ..., f(\beta_i, 2^m-1, m)].\end{cases}$$

Then in the multi-step prediction case, the minimum sum of control benefit $\Delta\mathbb{E}\big(J_{\boldsymbol{m}}\big(x(t)\big), j\big)$ after j-th slot allocation can be calculated as

$$\Delta\mathbb{E}\big(J_{\boldsymbol{m}}\big(x(t)\big), j\big) = \min_i \Big(\Delta\mathbb{E}\big(J_{\boldsymbol{m}}\big(x(t)\big), j-1\big) + \Delta\mathbb{E}\big(J_{\boldsymbol{m}}\big(x_i(t)\big)\big)\Big), \tag{18}$$

Algorithm 2: ADPS-m Algorithm

```
Input: $\Theta_i, L, N, x(t), m, i \in \{1, 2, ..., N\}$.
Output: $S(t+1)$.
calculate $2^m - 2$ predicted states and $\Lambda_m(t+1)$ in Eq. (16);
$S(t+1) = \varnothing$, $\Delta\mathbb{E}\big(J_{\boldsymbol{m}}\big(x(t+1)\big), 1\big) = +\infty$;
$\Phi_m(1) \leftarrow \Lambda_m(t+1)$;
for all $j \in \{1, 2, ..., L\}$ do
    for all $i \in \{1, 2, ..., N\}$ do
        $\Delta\mathbb{E}\big(J_{\boldsymbol{m}}\big(x_i(t)\big)\big) = \sum(\Phi_{m,i}(j) \odot \Theta_i - \Phi_{m,i}(j))$;
    end
    for all $i \in \{1, 2, ..., N\}$ do
        $\Delta\mathbb{E}\big(J_{\boldsymbol{m}}\big(x(t)\big), j\big) \leftarrow \min_i \Big(\Delta\mathbb{E}\big(J_{\boldsymbol{m}}\big(x(t)\big), j\big), \Delta\mathbb{E}\big(J_{\boldsymbol{m}}\big(x(t)\big), j-1\big) + \Delta\mathbb{E}\big(J_{\boldsymbol{m}}\big(x_i(t)\big)\big)\Big)$,
    end
    $i_j^* \leftarrow i$ with the minimum $\Delta\mathbb{E}\big(J_{\boldsymbol{m}}\big(x(t)\big), j\big)$;
    update $\Phi_{m,i}$: $\Phi_{\boldsymbol{m}}(j+1) \leftarrow \begin{bmatrix} 1 & ... & i_j^* & ... & N \\ \Phi_{m,1}(j) & ... & \Phi_{m,i_j^*}(j) \odot \Theta_{i_j^*} & ... & \Phi_{m,N}(j) \end{bmatrix}$;
    $S(t+1) \leftarrow [S(t+1) \quad i_j^*]$.
end
return $S(t+1)$
```

Comparing Eq. (18) with Eq. (10), the scheduling process of ADPS-m is the same as ADPS as shown in Eqs. (9) and (11) in computing the maximum gradient in Eq. (17). Performing m-step prediction calculate $2^{m+1} - 2$ physical states for each loop. The complexity of ADPS-m is $O\big(N(2L + 2^{m+1} - 2)\big)$. As indicated in Sec. 8.2.1, the preferable m is not large, since the expected overall control cost (16) will be inaccurately predicted as m increases. Hence the exponential term 2^{m+1} is an approximately constant term. In Sec. 8.2.1, we evaluate the execution time and control performance of ADPS-m under different prediction steps m, and the results reflect the trade-off between computation time and control performance.

6 SCHEDULING HETEROGENEOUS DATA FLOWS: ADPS-H/ADPS-H-M

Finally, we present the scheduling for heterogeneous data flows with two heterogeneous configurations – different data sizes and different control frequencies. Existing works, e.g., [7, 19, 21, 29, 41, 45] assume that the delivery of the uplink or downlink data stream can be done within one slot. However, due to the limited slot length in low-power wireless networks and the rich information contained in one message for sophisticated physical processes, a single sensing or control data may be composed of multiple transmission packets [46]. In the IEEE 802.15.4 standard [1], *aBaseSlotDuration*, which is the number of bytes forming a superframe slot, is 60. If the control command is a vector with a dimension larger than 8 and applies *double* data types (8 bytes) for each control value, it cannot be transmitted within one time slot. Moreover, the model predictive controller always generates the control commands for a given horizon (3-5 times of the physical state dimension) [27]. In this case, the transmission of multiple control vectors within a single time slot is even impossible. For heterogeneous sensing/control tasks and varying underlying processes in the multi-loop control system, the data sizes for different processes are often different [44]. Moreover, different loops may have different control requirements. Some loops may require more frequent control. Meanwhile, the limited bandwidth of the wireless network may not accommodate the scheduling of control commands for all loops at the same time. In our ADPS-H/ADPS-H-m, we assume that the control periods are predefined according to the control requirements and network bandwidth. It is worth noting that our scheduling strategy is different from the static [38] and dynamic [9] period assignment or packet size adaptation [15], the ADPS-H/ ADPS-H-m are proposed for scheduling the heterogeneous data flows with any given control periods and data sizes.

We set the coordination period T_c as the greatest common divisor (GCD) of the control periods of the N control loops, i.e., $T_c = \mathrm{GCD}(\mathcal{T})$. Each control period is an integer multiple of the superframe duration, and each superframe duration is an integer multiple of the time slot duration. Hence, the GCD of the control periods is an integer multiple of the time slot duration. Given $T_i = k_i T_c$, $k_i \in \mathbb{N}^+$, some loops may be skipped in the coordination period. We consider these skipped loops as 0 time slot occupation, i.e.,

$$\bar{H}_i = \begin{cases} 0, & \text{if } (t+1) \bmod T_i/T_c \neq 0 \\ H_i. & \text{else,} \end{cases} \tag{19}$$

While $\bar{H}_i = 0$, the scheduler will not allocate time slots in the next superframe to those loops. Unlike ADPS-m, scheduling heterogeneous flows that operate at different T_i makes the prediction steps m_i not identical with each other within the same prediction interval. Assuming that the minimum control period the GCD of N control periods, i.e., $T_{min} = T_c$, hence, we refer the overall prediction step m of the heterogeneous loops to the “fastest” control loop which has the minimum control period T_{min}, i.e., $m = m_{Tmin}$. Then the prediction steps of other loops can be determined by Eq. (20).

$$m_i = \left\lceil \frac{m_{Tmin} T_{min}}{T_i} \right\rceil. \tag{20}$$

Consider scheduling based on the general multi-step predictions, the control cost for the heterogeneous data flow of loop i can be expressed as

$$\begin{aligned} \mathbb{E}\Big(\mathcal{J}_i\big(x_i(t), \beta_i(t), T_i, m_i, \bar{H}_i\big)\Big) &= \mathbb{E}\Big(J_{H_i,m_i}\big(x_i(t)\big)\Big) = \sum_{r=1}^{m_i} \alpha^{r-1} \mathbb{E}\Big(J\big(x_i(t+r)\big)\Big) \\ &= \sum_{r=1}^{m_i} \alpha^{r-1} \sum_{l=1}^{2r} p_{l,i}(t+r) J\big(x_{l,i}(t+r)\big) \\ &= \sum_{r=1}^{m_i} \sum_{g=0}^{2^r-1} \lambda_{g,r,i} f(\mathcal{B}_i, g, r)^{n_i(t+1)}, \end{aligned} \tag{21}$$

where $\mathcal{B}_i(t)$ is the failure ratio for a single transmission of loop i in the heterogeneous case, i.e., $\mathcal{B}_i(t) = 1 - (1 - \beta_i(t))^{\bar{H}_i}$, and $\lambda_{g,r,i}$ and $f(\mathcal{B}_i, g, r)$ are the same with those in Sec. 5. Then the total control cost for the heterogeneous data flows can be expressed as

$$\mathbb{E}\Big(\mathcal{J}\big(x(t), \beta(t), \mathcal{T}, m, \mathcal{H}\big)\Big) = \mathbb{E}\Big(J_{\boldsymbol{H},\boldsymbol{m}}\big(x(t)\big)\Big) = \Lambda_{Hm}(t)\Theta_{Hm}^T(t) = \sum_{i=1}^{N} \sum_{r=1}^{m_i} \sum_{g=0}^{2^r-1} \lambda_{g,r,i} f(\mathcal{B}_i, g, r)^{n_i(t+1)}, \tag{22}$$

where

$$\begin{cases}\Lambda_{Hm}(t) & = \left[\Lambda_{m_1,1}(t), ..., \Lambda_{m_N,N}(t)\right] = \\ & \left[[\lambda_{0,1,1}, \lambda_{1,1,1}, ..., \lambda_{0,m_1,1}, ..., \lambda_{2^{m_1}-1,m_1,1}], ..., [\lambda_{0,1,N}, \lambda_{1,1,N}, ..., \lambda_{0,m_N,N}, ..., \lambda_{2^{m_N}-1,m_N,N}]\right] \\ \Theta_{Hm}(t) & = \left[\Theta_{m_1,1}(t), ..., \Theta_{m_N,N}(t)\right] = \\ & \left[[1, f(\mathcal{B}_1, 1, 1)^{n_1(t+1)}, ..., 1, ..., f(\mathcal{B}_1, 2^{m_1} - 1, m_1)^{n_1(t+1)}],\right. \\ & \left. ..., [1, f(\mathcal{B}_N, 1, 1)^{n_N(t+1)}, ..., 1, ..., f(\mathcal{B}_N, 2^{m_N} - 1, m_N)^{n_N(t+1)}]\right].\end{cases}$$

The benefit of the control cost for loop i at j-th slot (occupied slot $j - \bar{H}_i + 1$ to j-th) can be calculated as

$$\Delta\mathbb{E}\Big(J_{H_i,m_i}(x_i(t)), j\Big) = \sum \left(\Phi_{H_i,m_i,i}(j) \odot \Theta_i - \Phi_{H_i,m_i,i}(j)\right) = \sum_{r=1}^{m_i} \sum_{g=0}^{2^r-1} \lambda_{g,r,i} f(\mathcal{B}_i, g, r)^{n_i(t+1,j)} (f(\mathcal{B}_i, g, r) - 1), \tag{23}$$

where

$$\begin{cases}\Phi_{H_i,m_i,i}(j) = [\lambda_{0,1,i}, \lambda_{1,1,i} f(\mathcal{B}_i, 1, 1)^{n_i(t+1,j)}, ..., \lambda_{0,m_i,i}, ..., \lambda_{2^{m_i}-1,m_i,i} f(\mathcal{B}_i, 2^{m_i} - 1, m_i)^{n_i(t+1,j)}] \\ \Theta_i = [1, f(\mathcal{B}_i, 1, 1), ..., 1, ..., f(\mathcal{B}_i, 2^{m_i} - 1, m_i)].\end{cases}$$

The minimum sum of control benefit $\Delta\mathbb{E}\big(J_{\boldsymbol{H},\boldsymbol{m}}(x(t)), j\big)$ after j-th slot allocation is

$$\Delta\mathbb{E}\big(J_{\boldsymbol{H},\boldsymbol{m}}(x(t)), j\big) = \min_i \Big(\Delta\mathbb{E}\big(J_{\boldsymbol{H},\boldsymbol{m}}(x(t)), j - H_i\big) + \Delta\mathbb{E}\big(J_{H_i,m_i}(x_i(t))\big)\Big). \tag{24}$$

In contrast to ADPS and ADPS-m, ADPS-H and ADPS-H-m not only compute the update of $\Phi_{H_{i_j^*},m_{i_j^*},i_j^*}(j+1)$ for the optimal gradient at time slot j, but also update the $\{\Phi_{H_i,m_i,i}(j - \bar{H}_{i_j^*} + 1), ..., \Phi_{H_i,m_i,i}(j-1)\}$ for the former optimal gradients during the time slot $j - \bar{H}_{i_j^*} + 1$ to $j - 1$ because the control benefit (24) can achieve optimal at the j-th slot allocation if the time slots from $j - \bar{H}_{i_j^*} + 1$ to j are reallocated to loop i_j^*. Specifically, we define d_i as the difference in the number of transmissions for loop i between the scheduling results after $j-1$ slots allocation $S(t+1, j-1)$ and $S(t+1, j)$ after j slots allocation, i.e., $d_i = |i \in S(t+1, j)| - |i \in S(t+1, j-1)|$, where $|i \in S(t+1, j)|$ denotes the number of i that appears in the first j elements of $S(t+1)$. Then the update of $\Phi_{H_i,m_i,i}$ can be expressed as

$$\Phi_{H_i,m_i,i}(j+1) = \Phi_{H_i,m_i,i}(j) \odot \Theta_i^{d_i}. \tag{25}$$

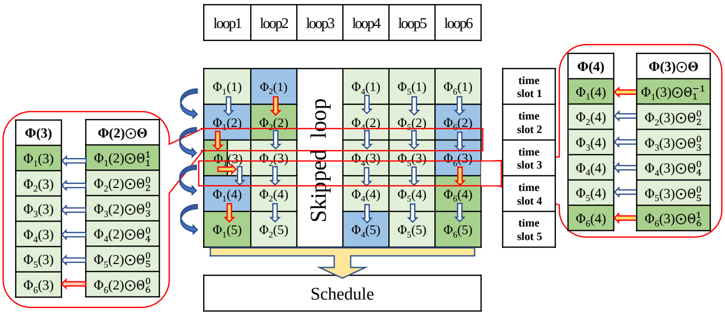


Fig. 5. The diagram of ADPS-H at time t when $\bar{H}_1 = \bar{H}_2 = \bar{H}_3 = \bar{H}_4 = 1, \bar{H}_3 = 0, \bar{H}_6 = 2$.

This way, ADPS/ ADPS-m is extended to ADPS-H/ ADPS-H-m for heterogeneous data flows. Fig. 5 shows an example of slot allocation for ADPS-H. For simplicity, the subscripts m_i as well as H_i of $\Phi_{m_i,H_i,i}$ are omitted. When slot 2 is assigned for the first time, the optimal gradient is $i_2 = 1$, and $\Phi_1(3)$ is updated by Eq. (24) ($d_1 = 1$). While scheduling slot 3, the $\Delta\mathbb{E}\big(J_{\boldsymbol{H},\boldsymbol{m}}(x(t)), 3\big)$ is minimized when slots 2 and 3 are assigned to loop 6 together. Then remove the transmission scheduling that originally assigned time slot 2 to loop 1 and update $\Phi_1(3)$ as well as $\Phi_6(4)$ ($d_1 = -1, d_6 = 1$). Since the optimal gradient i_j^* corresponding to any slot j can be successfully scheduled, we can achieve the minimum value of control cost (22). ADPS-H still features the low computational complexity of ADPS. As shown in Alg. 3, when the prediction step $m = 1$, ADPS-H-m is equivalent to ADPS-H. It is not hard to see that the computational complexity of the ADPS-H algorithm is $O\big(N(3L+2)\big)$. The computational complexity of the ADPS-H-m is $O\big(N(3L + 2^{m+1} - 2)\big)$. Again, as illustrated in Sec. 4, the exponential term 2^{m+1} is an approximately constant term since m is not large.

Lemma 6.1. *Assuming that the packet losses are independent and identically distributed (i.i.d.) and the link failure ratio β_i remains invariant within one superframe, ADPS-H-m provides the optimal solution to the problem* (6).

Proof. The proof of Lemma 6.1 is given in Appendix A.1. □

Algorithm 3: ADPS-H-m Algorithm

```
Input: $\Theta_i, L, N, x(t), \bar{H}_i, m, i \in \{1, 2, ..., N\}$.
Output: $S(t+1)$.
calculate $m_i$, the $m_i$-step predicted states of all loops, and $\Lambda_{Hm}(t+1)$ in Eq. (22);
$S(t+1, 1) = \varnothing$, $\Delta\mathbb{E}\big(J_{\boldsymbol{H},\boldsymbol{m}}(x(t)), 1\big) = +\infty$;
$\Phi_{Hm}(1) \leftarrow \Lambda_{Hm}(t+1)$;
for $j \le L$ do
    for all $i \in \{1, 2, ..., N\}$ do
        $\Delta\mathbb{E}\big(J_{H_i,m_i}(x_i(t))\big) = \sum \big(\Phi_{H_i,m_i,i}(j) \odot \Theta_i - \Phi_{H_i,m_i,i}(j)\big)$;
    end
    for all $i \in \{1, 2, ..., N\}$ do
        $\Delta\mathbb{E}\Big(J_{\boldsymbol{H},\boldsymbol{m}}\big(x(t)\big), j\Big) \leftarrow$
          $\min_i \Big(\Delta\mathbb{E}\big(J_{\boldsymbol{H},\boldsymbol{m}}(x(t)), j\big), \Delta\mathbb{E}\big(J_{\boldsymbol{H},\boldsymbol{m}}(x(t)), j - H_i\big) + \Delta\mathbb{E}\big(J_{H_i,m_i}(x_i(t))\big)\Big)$;
    end
    for all $i \in \{1, 2, ..., N\}$ do
        $d_i = |i \in S(t+1, j)| - |i \in S(t+1, j-1)|$;
    end
    $i_j^* \leftarrow i$ with the minimum $\Delta\mathbb{E}\big(J_{\boldsymbol{H},\boldsymbol{m}}(x(t)), j\big)$;
    form the $S(t+1, j)$;
    update $\Phi_{H_i,m_i,i}$: $\Phi_{\boldsymbol{H},\boldsymbol{m}}(j+1) \leftarrow$
      $\begin{bmatrix} 1 & \dots & i_j^* & \dots & N \\ \Phi_{H_1,m_1,1}(j) \odot \Theta_1^{d_1} & \dots & \Phi_{H_{i_j^*},m_{i_j^*},i_j^*}(j) \odot \Theta_{i_j^*}^{d_{i_j^*}} & \dots & \Phi_{H_N,m_N,N}(j) \odot \Theta_N^{d_N} \end{bmatrix}$
    if $j = L$ then
        $S(t+1) \leftarrow S(t+1, L)$;
    end
end
return $S(t+1)$.
```

Remark 6.1. *Currently, we set the coordination period T_c as the GCD of the control periods of N loops to guarantee the optimality and the stability of the multi-loop networked control system. If $T_c > GCD(\mathcal{T})$, loops whose control periods $T_i < T_c$ would reuse the schedule which is calculated at the* *beginning of the current coordination period until the next T_c begins, thus the optimality of scheduling cannot be guaranteed. Moreover, according to* (32) *to* (34)*, the stability condition in Sec. 7.2 is only valid for the optimal solution.*

Finally, we establish a new optimization problem (26) which derives the optimal scheduling results $\{S(t+1), ..., S(t+m)\}^*$ for a longer T_c. The longer T_c can be set to the least common multiple of the control periods, e.g., $T_c = \text{LCM}(\mathcal{T})$. The optimal scheduling result aims at minimizing the control costs over a longer time horizon by $\{n_i(\tau)\}_{\tau=t+1}^{t+m}$, which is only based on the predicted physical states at the beginning of the coordination period and applied in sequence for the long horizon. However, our optimal scheduling (6) solving the control cost at a particular time step based on the real-time states in each control periods. In addition, the increase in coordination period prevents loops with low control periods from responding to physical disturbance timely since the schedule is only updated at the beginning of the prediction interval. In Sec. 8, we carefully evaluate the control performance of the two choices to support our scheduling strategy.

$$\begin{aligned}\min_{\{n_i(\tau)\}_{\tau=t+1}^{t+m}} \quad & \mathbb{E}\Big(\mathcal{J}\big(x(t), \beta(t), \mathcal{T}, m, \mathcal{H}\big)\Big) \\ s.t. \quad & \sum_{i=1}^{N} H_i n_i(\tau) \leq L \\ & n_i(\tau) \in \{0, 1, ..., L\} \\ & \forall i \in \{1, 2, ..., N\}, \tau \in \{t+1, ..., t+m\}\end{aligned} \tag{26}$$

7 STABILITY ANALYSIS

In this section, we establish a sufficient condition of the mean-square stability for ADPS-H-m. As illustrated in Sec. 3, ADPS, ADPS-m, and ADPS-H are all special cases of ADPS-H-m, if the multi-loop control system can be stable under ADPS-H-m, then the stability of which is scheduled by the above special cases can also be guaranteed.

DEFINITION 1. *[8] A discrete-time stochastic system is mean-square stable (MSS) if any initial state $x(0)$ satisfies*

$$\limsup_{t\to\infty} \mathbb{E}\left(x^T(t)x(t)\right) = 0.$$

A closed-loop system is called MSS if there exists a stochastic Lyapunov function $V(x)$ satisfying the following conditions.

1) $V(0) = 0$ and $V(x) > 0, \forall x \neq 0$.

2) $\|x\| \to \infty \Rightarrow V(x) \to \infty$.

3) $\mathbb{E}\big(V(x)\big)$ decreases along system trajectories, i.e.,

$$\mathbb{E}\Big(V\big(x(t+1)\big)\Big) - \mathbb{E}\Big(V\big(x(t)\big)\Big) \leq 0. \tag{27}$$

We show that the ADPS-H-m algorithm can guarantee the MSS of the multi-loop control system under the assumption: the existence of any fixed periodic schedule such that the resultant system is MSS.

We first need to determine whether there is a fixed periodic schedule that makes the closed-loop system MSS. We conclude the conditions that a fixed periodic schedule should satisfy to make the closed-loop MSS in Sec. 7.1. Then in Sec. 7.2, we analyze the MSS of the multi-loop control system.

7.1 MSS Analysis of DT-LTI System with Fixed Periodic Schedule

Consider a multi-loop DT-LTI system, with system dynamics of the loop i being given by Eq. (1). Assume that the state feedback gain K_i renders each loop i asymptotically stable in the ideal network without packet loss. Let a binary variable $\varphi_i(t)$ denote end-to-end packet reception ($\varphi_i(t) = 1$) or loss ($\varphi_i(t) = 0$). According to [7], the dynamics of loop i are equivalent to the augmented system:

$$z_i(t+1) = \tilde{A}_i(s,t) z_i(t) \tag{28}$$

where $z_i(t) = \left[x_i^T(t)\ \hat{u}_i^T(t)\ u_i^T(t)\right]^T$ and

$$\tilde{A}_i(s,t) = \begin{bmatrix} A_i & B_i & 0 \\ 0 & 1 - \varphi_i(s) & \varphi_i(s) \\ K_i A_i & 0 & K_i B_i \end{bmatrix}$$

1) if $\varphi_i(t) = 1$

$$\hat{u}_i(t) = u_i(t), z_i(t+1) = \hat{z}_i^c(t+1) = \tilde{A}_i^c z_i(t).$$

2) if $\varphi_i(t) = 0$, $\hat{u}_i(t-1)$ is adopted and

$$\hat{u}(t) = \hat{u}_i(t-1), z_i(t+1) = \hat{z}_i^o(t+1) = \tilde{A}_i^o z_i(t).$$

Here,

$$\tilde{A}_i^c = \begin{bmatrix} A_i & B_i & 0 \\ 0 & 0 & 1 \\ K_i A_i & 0 & K_i B_i \end{bmatrix}, \tilde{A}_i^o = \begin{bmatrix} A_i & B_i & 0 \\ 0 & 1 & 0 \\ K_i A_i & 0 & K_i B_i \end{bmatrix}.$$

Under Assumption 3.1, the $\tilde{A}_i(s,t)$ can be rewritten as [7, 29]

$$\tilde{A}_i(s,t) = \tilde{A}_{0i} + \tilde{A}_{1i} p_i(t), \tag{29}$$

where $p_i(t)$ are i.i.d random variables with $\mathbb{E}(p_i(t)) = 0$, $\mathrm{Var}(p_i(t)) = \sigma_{p_i}^2$, and $\mathbb{E}(p_i(t)p_j(t)) = 0, \forall i, j \in \{1, 2, ..., N\}$,

$$\tilde{A}_{0i} = \begin{bmatrix} A_i & B_i & 0 \\ 0 & (1 - \mu_{\varphi_i}(s))I & \mu_{\varphi_i}(s)I \\ K_i A_i & 0 & K_i B_i \end{bmatrix},$$

$$\tilde{A}_{1i} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mu_{\varphi_i}(s)I & -\mu_{\varphi_i}(s)I \\ 0 & 0 & 0 \end{bmatrix}, \sigma_{p_i}^2(s) = \frac{1}{\mu_{\varphi_i}(s)} - 1.$$

As the packet losses of each link are i.i.d according to Assumption 3.1. The end-to-end packet delivery variables $\{\varphi_i(t), t \in \mathbb{N}\}$ under the fixed periodic schedule with a certain transmission number n_i are i.i.d with an average of μ_{φ_i} since the product of mutually independent Bernoulli random variables is also a Bernoulli random variable [25]. $p_i(t) = 1 - (\varphi_i(t)/\mu_{\varphi_i})$ is a binary random variable that takes 1 or $1 - (1/\mu_{\varphi_i})$ with $\mathbb{P}(p_i(t) = 1) = 1 - \mu_{\varphi_i}$ and $\mathbb{P}(p_i(t) = 1 - 1/\mu_{\varphi_i}) = \mu_{\varphi_i}$. For the discrete-time stochastic system, Lemma 7.1 gives a condition to check whether the system is MSS.

Lemma 7.1. *[8, p.131] System* (28) *is MSS if and only if there exists a positive definite matrix* P_i *satisfying*

$$\tilde{A}_{0i}^T P_i \tilde{A}_{0i} - P_i + \sigma_{p_i}^2 \tilde{A}_{1i}^T P_i \tilde{A}_{1i} < 0 \tag{30}$$

Theorem 7.2. *If there exists a positive definite matrix P_i satisfying* (30) *for each loop i, $\forall i \in \{1, 2, ..., N\}$, the multi-loop control system described by Eq.* (31) *that contains N loops is mean square stable under the fixed periodic schedule S_f.*

$$z(t+1) = \tilde{A}(s,t)z(t) \tag{31}$$

where

$$\tilde{A}(s,t) = \text{blkdiag}\big(\tilde{A}_1(s,t), \tilde{A}_2(s,t), ..., \tilde{A}_N(s,t)\big),$$
$$z(t) = [z_i(t), z_2(t), ..., z_N(t)]^T.$$

Proof. According to the LMI condition in Lemma. 7.1, if each loop i can derive its own positive definite matrix P_i and verify the MSS, then the multi-loop control system Eq. (31) can also be MSS as there exists a positive definite matrix $P = \text{blkdiag}(P_1, P_2, ..., P_N)$ which satisfies (30). □

7.2 Stability Condition of ADPS-H-m

Given the existence of a fixed periodic schedule that guarantees the overall system MSS, we can establish that the closed-loop system with the ADPS-H-m algorithm is also MSS. First, we need to prove that the objective function $J_{H,m}(x(t))$ is a stochastic Lyapunov function of the overall multi-loop control system under the fixed periodic schedule S_f in Sec. 7.1.

Lemma 7.3. *$J_{H,m}(x(t))$ is a stochastic Lyapunov function of the overall control system under the fixed periodic schedule S_f in Sec. 7.1.*

Proof. The proof of Lemma 7.3 is given in Appendix A.2. □

Proposition 1. *If there exists a fixed periodic schedule S_f such that the overall system with N loops is MSS and $J_{H,m}(x(t))$ is a stochastic Lyapunov function with S_f, then the closed-loop system with the optimal schedule S^* derived by solving* (6) *is also MSS.*

Proof. Since $J_{H,m}(x(t))$ is a stochastic Lyapunov function of the closed-loop system resulting from a fixed periodic schedule S_f according to Lemma 7.3, we denote the value of the stochastic Lyapunov function of the fixed periodic schedule S_f as $J'_{H,m}(x(t))$, and the value of the optimal schedule S^* as $J^*_{H,m}(x(t))$. Therefore,

$$\mathbb{E}\Big(J'_{H,m}(x(t+1))\Big) \leq \mathbb{E}\Big(J^*_{H,m}(x(t))\Big). \tag{32}$$

Because the schedule S^* by ADPS-H-m algorithm minimizes the objective function $\mathbb{E}(J_{H,m}(x(t)))$, we have

$$\mathbb{E}\Big(J^*_{H,m}(x(t+1))\Big) \leq \mathbb{E}\Big(J'_{H,m}(x(t+1))\Big). \tag{33}$$

Combining (32) and (33), we can derive

$$\mathbb{E}\Big(J^*_{H,m}(x(t+1))\Big) \leq \mathbb{E}\Big(J^*_{H,m}(x(t))\Big). \tag{34}$$

For the optimally scheduled system, $\mathbb{E}(J_{H,m}(x(t)))$ decreases along the trajectories of the system and also satisfies $J_{H,m}(x(t)) > 0, \forall x \neq 0$, and $J_{H,m}(x(t)) \longrightarrow \infty$ as $\|x\| \longrightarrow \infty$. Therefore, $J_{H,m}(x(t))$ is also a stochastic Lyapunov function of the optimally scheduled system by ADPS-H-m. □

8 EVALUATION

In this section, systematic experiments are performed to evaluate the performance of the proposed ADPS, ADPS-m, ADPS-H, and ADPS-H-m strategies.

8.1 Experimental Setup

The scheduling strategies are tested on four double water-tank systems, which share a wireless network. The physical systems are simulated with MATLAB/Simulink [29]. Models all run at the frequency of 960 Hz in Simulink. All control commands are updated at 24 Hz ($T_i = 41.6$ ms) for the evaluation of ADPS and ADPS-m. In heterogeneous flow simulations, we modify the control period T_i of loops 3 and 4 to $T_3 = 83.3$ ms (12 Hz) and $T_4 = 166.6$ ms (6 Hz).

8.1.1 Multi-loop Control System. The three-state double water-tank systems which are modeled as (35) are discretized by Euler's method [6, 26], and state feedback controllers are deployed for reference tracking.

$$\begin{aligned}\dot{L}_1 =& \frac{1}{\rho A_1}\Big(\alpha u - \frac{\sqrt{\rho g}}{\rho R_1}\sqrt{L_1}\Big)\\ \dot{L}_2 =& \frac{1}{\rho A_2}\Big(\frac{\sqrt{\rho g}}{\rho R_1}\sqrt{L_1} - \frac{\sqrt{\rho g}}{\rho R_2}\sqrt{L_2}\Big)\\ \dot{L}_R =& \frac{1}{\rho A_R}\Big(\frac{\sqrt{\rho g}}{\rho A_R}\sqrt{L_2} - \alpha u\Big).\end{aligned} \tag{35}$$

The control performance is evaluated in terms of mean absolute error (MAE) in Eq. (36),

$$\text{MAE} = \frac{1}{D+1}\sum_{t=0}^{D}|x(t) - x_{ref}(t)|, \tag{36}$$

where D is the total number of control periods and x_{ref} is the reference state of water tank. We set up 4 sets of water-tank systems, where systems 1 and 3 share the same configuration parameters and systems 2 and 4 use another configuration as shown in Tab. 1. The state feedback gains are $K_1 = K_3 = [-5\ \ -20]$, $K_2 = K_4 = [-60\ \ -80]$ for ADPS and ADPS-m. In heterogeneous flow simulations, the state feedback gains for loops 3 and 4 are updated to $K_3 = [-10\ \ -4]$ and $K_4 = [-50\ \ -20]$, respectively, since the control periods are adjusted.

Table 1. System Parameters

System 1 & 3				System 2 & 4			
par	value	par	value	par	value	par	value
A_1	0.01	R_1	0.0006	A_1	0.12	R_1	0.0006
A_2	0.006	R_2	0.0008	A_2	0.007	R_2	0.0008
A_R	1	α	10	A_R	1	α	10

8.1.2 Wireless Network. We simulate IEEE 802.15.4 beacon-enabled wireless network. The wireless traces are collected from four links of the WSAN testbed at Washington University, which contains the connectivity of links and Received Signal Strength Indication (RSSI). By feeding in the traces of received signal strengths and noise strengths from the testbed to TOSSIM [11], a standard simulator of TinyOS with a realistic link model [24], we can evaluate the wireless control systems under a wide range of wireless conditions. We introduce offsets to the noise traces to simulate varying levels of noise. By varying the offsets in different experiments, we can simulate network conditions ranging from normal conditions to stress tests with excessive packet drops. The noise levels of the downlinks vary from -82 to -74 dBm. Fig. 6 shows the PRRs of four links under controlled noise levels. The PRRs vary among links under the same noise strength since the RSSIs are different. The statistics of PRRs are based on all bidirectional links in the topology with 91000 transmissions per link. The noise level of -82 dBm results in a 15% link failure rate. The noise level of -74 dBm causes around 70% link failures when RSSI is -67 dBm, which represents extreme conditions such as adversarial jamming attacks. Each superframe has 5 time slots, while each time slot lasts 8.3

ms. Since the first time slot is allocated to the beacon frame, we have 4 available time slots to allocate to actuation flows. For heterogeneous flow simulations, u_4 would use 2 time slots for a single transmission while others still occupy one, i.e., $\mathcal{H} = \{1, 1, 1, 2\}$. Since this paper focuses on downstream scheduling, we fix the end-to-end packet loss ratio of all upstream flows to $\beta = 0.2$ for fair comparisons, and we do not schedule upstream sensing flows at run-time, which could follow static schedules [29].

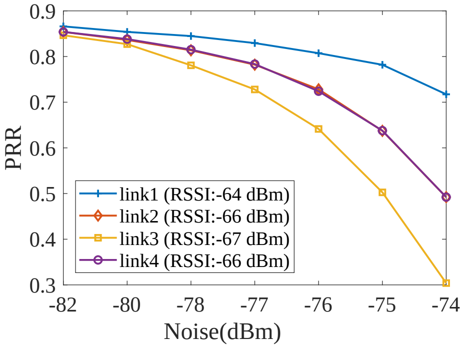


Fig. 6. PRRs under various noise levels.

8.2 Experiments and Evaluation

8.2.1 ADPS and ADPS-m under Cyber and Physical Disturbances. In a noisy environment, we run ADPS under several background noise levels. To better test the adaptability of the algorithm to physical disturbances, we add pulsed disturbance to loops 1 and 3 at $t = 4$ s and to loops 2 and 4 at $t = 9$ s.

We conduct four groups of experiments, which are ADPS, brute force (optimal), round-robin (RR), and an approach to enhance age-of-information (AoI) inspired from [43]. In each control period, the brute force method exhaustively traverses all schedule candidates to find the optimal $S^*(t+1)$. While under the RR, each loop is assigned time slots evenly. The AoI-based scheduler minimizes the expectation of AoI for N loops. According to the dynamics of AoI Δ_i of loop i in [43],

$$\Delta_i(t+1) = \begin{cases} \Delta_i(t) + 1, & \varphi_i(t+1) = 0, |i \in S(t+1)| \neq 0 \text{ or } |i \in S(t+1)| = 0 \\ 1, & \varphi_i(t+1) = 1, |i \in S(t+1)| \neq 0 \end{cases}$$

the expectation of AoI for the multi-loop system at time $t+1$ can be denoted as

$$\mathbb{E}\big(\Delta(t+1)\big) = \sum_{i=1}^{N} \Big((\Delta_i(t) + 1)(1 - \mathcal{B}^{n_i(t+1)}(t+1)) + \mathcal{B}^{n_i(t+1)}(t+1) \Big). \tag{37}$$

We finally minimize Eq. (37) by the dynamic allocation of time slots which is similar to (6), i,e.,

$$\begin{aligned} \min_{n_i(t+1)} \quad & \mathbb{E}\big(\Delta(t+1)\big) \\ s.t. \quad & \sum_{i=1}^{N} H_i n_i(t+1) \leq L \\ & n_i(t+1) \in \{0, 1, ..., L\} \\ & \forall i \in \{1, 2, ..., N\}. \end{aligned} \tag{38}$$

In the following experiments, we use the dynamic programming approach in the previous sections to solve (38). Simulations are performed under different constant background noise levels.

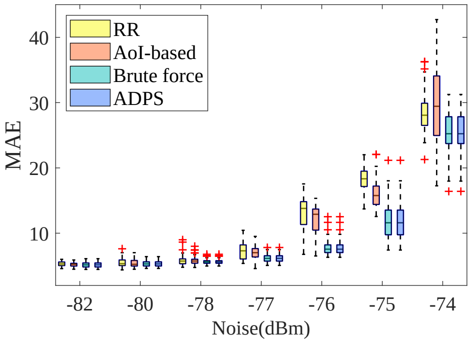


Fig. 7. MAEs affected by pulsed disturbance under noise levels from -82 to -74 dBm.

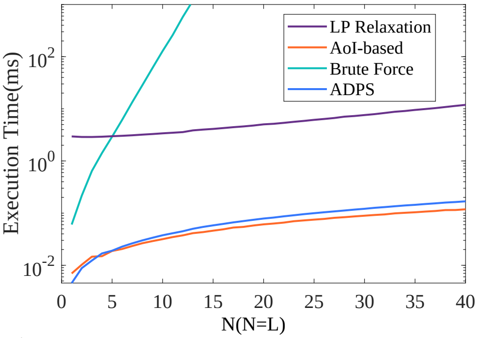


Fig. 8. Execution time of each algorithm under different numbers of loops and slots.

Using the network reception message obtained by TOSSIM, each boxplot of control performance is derived from 50 rounds of simulations with a simulation interval of 12 s. As shown in Fig. 7, in all noise levels, the ADPS achieves the same control performance as the brute force, which is better than those of both the RR and AoI-based scheduling, especially under stress tests with the noise levels from -76 to -74 dBm. When the network quality is good (noise level from -82 to -77 dBm), the MAEs of AoI-based scheduling are not distinguishable since retransmissions are not necessary and the network resource is sufficient. However, when the network quality is bad (noise level from -76 to -74 dBm), since the AoI-based scheduler does not schedule transmissions based on the online physical states when a physical disturbance occurs at run-time, the AoI-based scheduler may deteriorate the control performance due to its untimely response to the physical disturbance. The results in Fig. 7 show that just optimizing a network metric cannot achieve the specific benefit of scheduling according to the system dynamics on improving the control performance of the WNCS. When the noise level is -76 dBm, the average MAE of ADPS is 7.55, and ADPS improves the performance by 45.1% and 41.4% compared with RR (13.81) and the AoI-based approach (12.90), respectively. Under the level -75 dBm, the average MAE of ADPS is 11.60, which is 36.7% less than RR (18.34) and 26.5% less than the AoI-based approach (15.78). ADPS also has better performance than the RR and AoI-based approach under other noise levels, which implies the optimality and robustness of ADPS.

Fig. 8 shows the execution time statistics of ADPS compared with LP relaxation of the integer programming scheduling problem [29] with computational complexity of $O((N(L+1))^3 \ln(N(L+1)))$, the AoI-based approach, and brute force under different numbers of loops and slots. Each data point in Fig. 8 is obtained by averaging the latencies of 10000 times of algorithm execution on a 2.6 GHz Intel Core i7 processor. The LP problem is solved using `linprog` solver in MATLAB. The execution time of the brute force grows exponentially with N, while the ADPS grows approximately linear, and has significantly smaller computing latency compared with LP relaxation (less than 0.1 ms when $N \leq 15$). When $N = 4$, the average latency of ADPS is 0.014 ms, saving 99.5% execution time compared with LP relaxation (2.919 ms). By using the dynamic programming strategy to solve (38), the computational complexity of the AoI-based approach is $O(2NL)$, which can demonstrate that the dynamic programming-based approach can also be applied to other optimization problems that are similar to the form of (6). The AoI-based approach can save more execution time, however, Fig. 7 shows it cannot achieve the optimal control performance.

Moreover, we compare ADPS-m with ADPS across different noise levels. Fig. 9 shows the control performance of ADPS-m with $m = 3$ and $\alpha = 0.3$ compared with ADPS. Under the noise level of -76 dBm, the average MAE of ADPS-3 is 5.51, improving 27.0% control performance compared with ADPS (7.55). Under -75 and -74 dBm, the ADPS-3 improves the control performance by

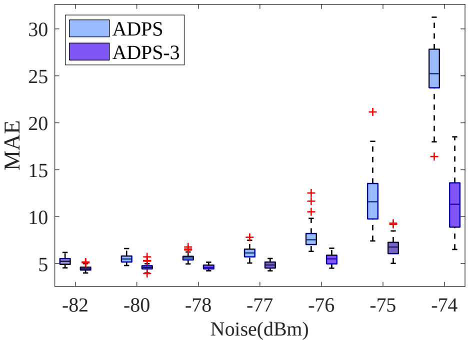
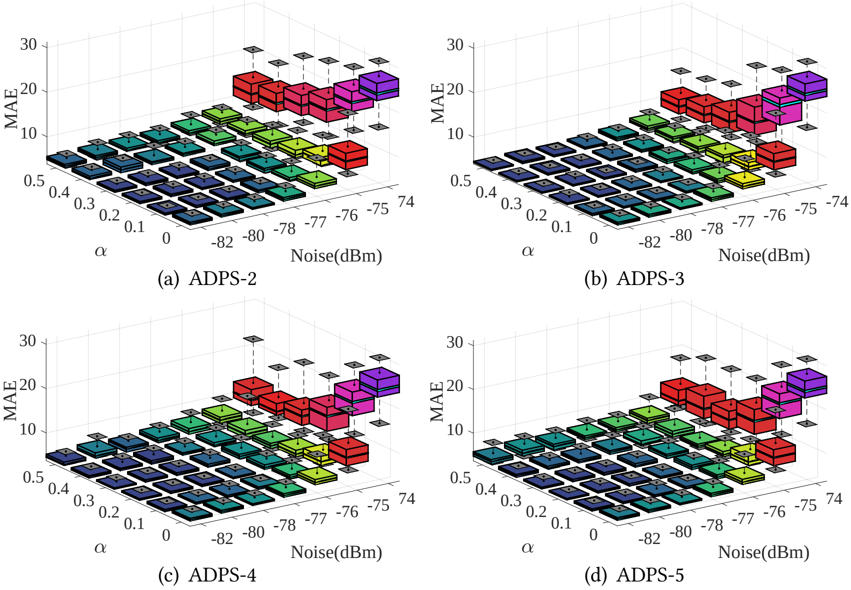


Fig. 9. MAEs of ADPS-m under constant background noise levels (m=3, α=0.3).

41.6% and 55.1% compared with ADPS, respectively. The results show the superiority of foresight prediction due to the consideration of future multi-step performance. Please note that this does not mean that ADPS is not optimal since ADPS and ADPS-m apply different scheduling objectives. In Figs. 10 and 13, we explore how the prediction step m and attenuation factor α affect the control performance and execution time (overhead). Note that when $\alpha = 0$, ADPS-m becomes ADPS with one-step prediction. Figs. 10(a) to 10(d) are the boxplots of MAE under different noise levels, α, and m. And we integrate the average values of them in Fig. 11 for a clearer comparison.



(a) ADPS-2

(b) ADPS-3

(c) ADPS-4

(d) ADPS-5

Fig. 10. MAEs of ADPS-m with different prediction steps m.

From Figs. 10(a) to 10(d), under the normal network noise levels from -82 to -76 dBm, the dark blue boxplot regions in each subplot indicate preferable control performance. While in the stress conditions of -75 and -74 dBm, the control performance of ADPS-m at any α and m is significantly improved compared with ADPS. This demonstrates the effectiveness of multi-step cyber and physical predictions. In Fig. 11, the ADPS-m algorithm with multi-step performance prediction outperforms the ADPS algorithm with one-step prediction in control performance at any noise level. Among the control performance surfaces corresponding to $m = 1$ to $m = 5$, ADPS-2 has obvious performance improvement compared with ADPS (ADPS-1). And ADPS-3, 4, 5 further improve the control performance compared with ADPS-2. Combining Figs. 10 and 12, the MAE

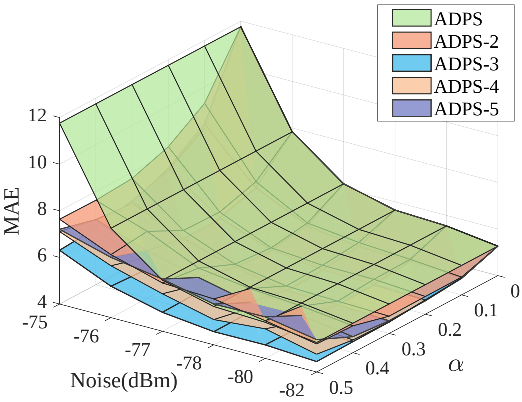
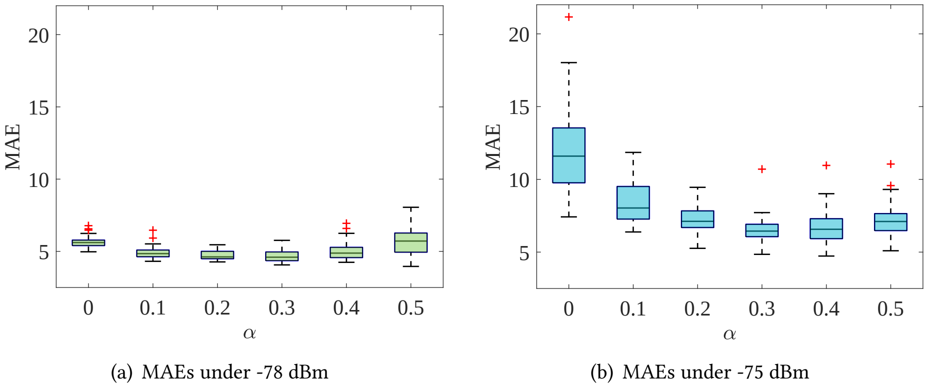


Fig. 11. Average MAEs of ADPS-m (m=2,3,4,5)

of ADPS-m algorithm for any m is decreasing monotonically as α increases from 0 to 0.3, and ADPS-3,4,5 achieve almost the same control performance with the same α when $\alpha \leq 0.3$. This is because the weight of performance prediction decreases exponentially in the overall expected control cost in Eq. (15) as m increases. However, the MAEs for ADPS-4 and ADPS-5 increase with α when $\alpha \geq 0.3$. This is mainly because the multi-step prediction error of PRR affects the accuracy of the performance prediction result, affecting the performance of the scheduling. Therefore, as shown in Fig. 11, the increase in m may not improve control performance but rather may cause a reduction due to the inaccuracy of the prediction. Hence, we choose $\alpha = 0.3$ and $m = 3$ for Sec. 8.2.2.



(a) MAEs under -78 dBm

(b) MAEs under -75 dBm

Fig. 12. MAEs of ADPS-5 under different α from 0 to 0.5.

In Fig. 13, we measure the execution time of ADPS-m under different m and compare them with brute force. It can be seen that the execution time of both ADPS-m and brute force increase monotonically as m increases. ADPS-m has significant advantages at any m. The execution time with $m = 3$ is less than 3 ms, which indicates its scalability.

8.2.2 *ADPS and ADPS-m under Variable Wireless Noise Levels.*

ADPS and ADPS-m are evaluated in the scenario with variable noise levels. In the first 5 s, the noise levels of loops 1 and 2 are -78 dBm, and those of loops 3 and 4 are -75 dBm. The background noise changes at $t = 5$ s. The noise levels of loops 1 and 2 increase to -75 dBm, and those of loops 3 and 4 decrease to -84 dBm. Figs. 14 and 15 show the comparative result of ADPS and ADPS-3 ($\alpha = 0.3$). In Fig. 14, we can see that the run-time absolute error corresponding to ADPS-3 has a faster convergence speed than ADPS during $t = 1$ to

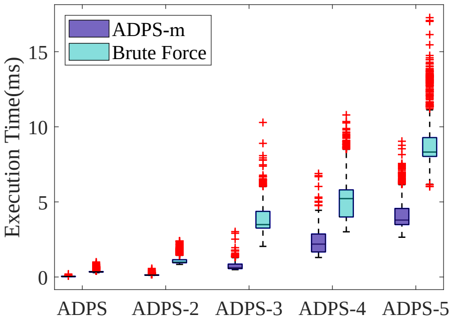


Fig. 13. Execution time of ADPS-m under different prediction steps *m*.

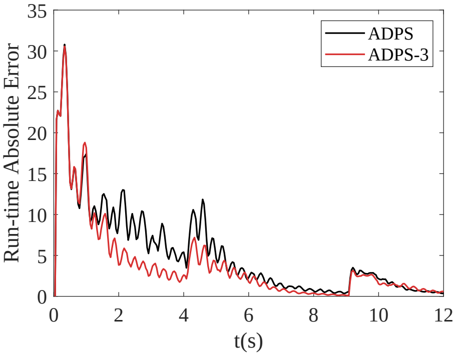


Fig. 14. Run-time absolute error under variable wireless noise levels.

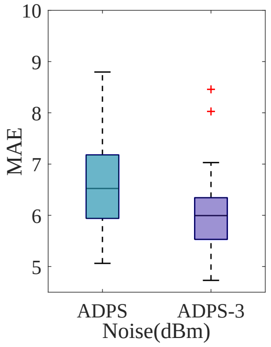


Fig. 15. MAEs of ADPS-m under constant background noise levels (m=3, α=0.3).

4 s. The MAEs under variable background noise levels are shown in Fig. 15, which demonstrates that ADPS and ADPS-m are able to adapt the variable noise levels and ADPS-m performs better than one-step prediction based ADPS.

8.2.3 *Simulation under Heterogeneous Data Flow Conditions.*

Finally, we test the effectiveness of the ADPS-H and ADPS-H-m strategies under heterogeneous data flows. The parameters of heterogeneous flows are introduced in Sec. 8.1. For comparison, we also conduct the RR scheduling and the AoI-based scheduling under heterogeneous flows.

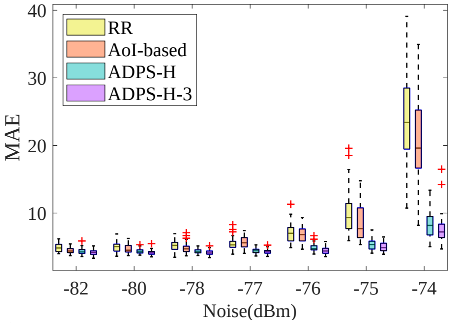


Fig. 16. MAEs under heterogeneous data flow conditions from the noise level -82 to -74 dBm.

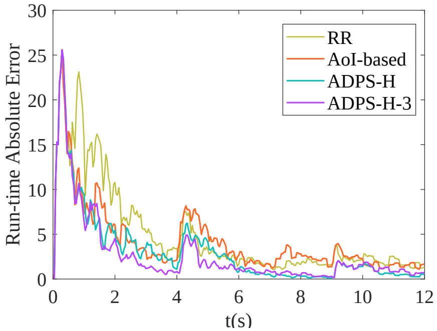


Fig. 17. Run-time absolute error of ADPS-H and ADPS-H-3 under -76 dBm.

Under constant noise levels, in Figs. 16 and 17, we can find that ADPS-H significantly outperforms the RR and the AoI-based approach in any noise levels. Under the noise level of -76 dBm, the average of MAE of ADPS-H is 4.74, improving 32.3% and 30.4% control performance compared with RR and the AoI-based scheduling, respectively. ADPS-H improves the control performance by 42.8% and 43.7% under -75 dBm, and 65.1% and 58.3% under -74 dBm, compared with RR and the AoI-based scheduling, respectively. The performance of AoI-based scheduler decreases mainly due to the untimely responses to the disturbances at $t = 4\ s$ and $9\ s$ shown in Fig. 17. Compared with ADPS-H, ADPS-H-m improves the control performance by 8.4%, 8.6%, and 12.0% under noise levels from -76 to -74 dBm, and gains faster convergence speed, since the limited network resources are allocated to loops that are more likely to improve the overall control performance in the finite horizon $m_i * T_i$, allowing the control commands at the actuators of these loops to be updated efficiently.

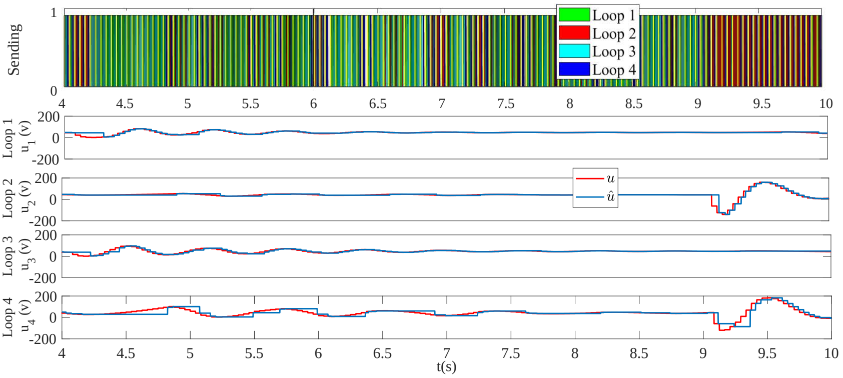


Fig. 18. Network scheduling and control command updating.

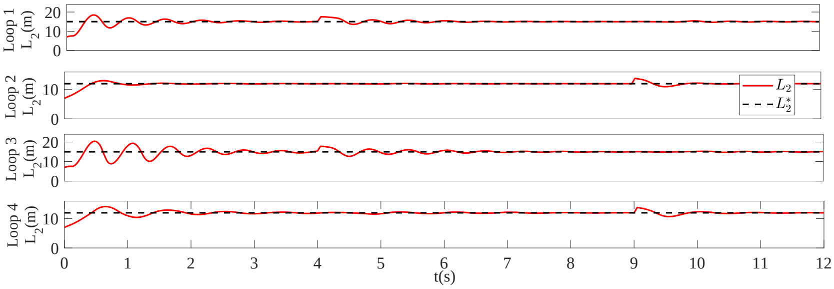


Fig. 19. Response curves of heterogeneous data flows, L_2^* is the reference state of L_2.

The detailed slot allocation results and corresponding control command updates during $t =$ 4 to 10 s under -76 dBm are presented in Fig. 18. And Fig. 19 presents the response curves corresponding to Fig. 18. As we can see, although loops 2 and 4 use the same physical parameters, loop 4 has longer control periods (6 Hz) and larger data size ($H_4 = 2$), thus the updating frequency of the control command in loop 4 is lower than that of loop 2 (blue solid lines). Combining Figs. 18 and 19, due to the pulse disturbance to loops 1 and 3 at time $t = 4$ s, ADPS-H-3 allocates more slots to schedule the control commands of loops 1 and 3 during $t =$ 4 to 7 s. While loops 2 and 4 are suffering disturbance at time $t = 9$ s, ADPS-H-3 provides more network resources for updating the control commands of loops 2 and 4.

Although the control period is not part of our design space, we also perform simulations with a comparative evaluation of our scheduling strategy when the control periods are reduced. Fig. 20 shows the control performance of the ADPS-H with previous control periods $\mathcal{T} = [41.6\ 41.6\ 83.3\ 166.6]$ ms, RR and ADPS-H with adjusted control periods $\mathcal{T}_a = [49.8\ 49.8\ 49.8\ 99.7]$ms denoted as RR-S and ADPS-H-S, respectively. Each superframe consists of 6 time slots and the number of available time slots is set to 5 (one for beacon frame). After increasing the transmission frequency of the control commands, the performance of RR-S is better than that of ADPS-H under the previous control periods, however, the ADPS-H-S still outperforms RR-S under the new adjusted control periods, which shows that our optimal scheduling strategy can improve the control performance of the wireless control system at any predefined control periods.

In Fig. 21, we evaluate the impact of coordination period on the control performance. The coordination periods for ADPS-H and ADPS-H-L are $T_c = \text{GCD}(\mathcal{T})$ and $T_c' = \text{LCM}(\mathcal{T})$, respectively. The schedule results of ADPS-H-L are derived by solving problem (26). As we can see, the MAEs of ADPS-H are better than ADPS-H-S under any noise level and gain a 14% performance improvement in the noise level of -74 dBm. This is mainly because ADPS-H derives the schedule $S(t+1)$ according to the run-time cyber and physical states, while ADPS-H-L derives the schedule only by the physical state and its prediction at the beginning of the longer T_c. In such case, ADPS-H-L cannot timely respond to the physical disturbance. The impacts on the control performance are enlarged as T_c increases.

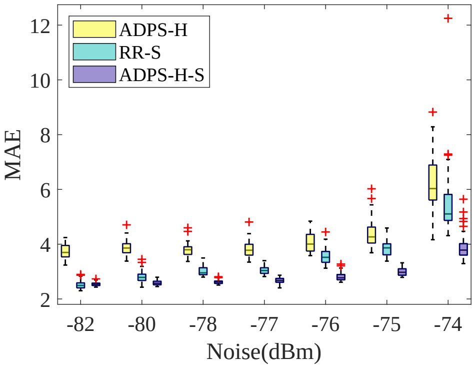


Fig. 20. MAEs of heterogeneous data flows with adjusted control periods $\mathcal{T}_a = [49.8\ 49.8\ 49.8\ 99.7]$ ms.

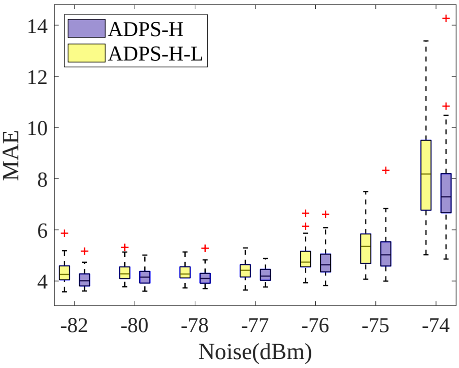


Fig. 21. MAEs of heterogeneous data flows with adjusted coordination period $T_c' = \text{LCM}(\mathcal{T}) = 166.6$ ms.

Table 2. Disturbance Settings

(a) Wireless Noise Level Settings

time	noise level(dBm)
0~6	-80
6~12	-76
12~18	-78
28~24	-75
24~30	-78

(b) Physical Disturbance Settings

loop	time(s)	loop	time(s)
1	5, 20, 25	6	10, 13, 25
2	5, 10, 15	7	10, 13, 25
3	5, 22, 25	8	5, 20, 25
4	10, 17, 25	9	10, 20, 25
5	10, 17, 25	10	5, 10, 25

Finally, we scale up the simulation of heterogeneous data flow scheduling from four loops to ten loops and evaluate the ADPS-H-m algorithm in the scenario with variable noise levels, the changes of noise levels and the physical disturbance settings are are shown in Tabs. 2(a) and 2(b), respectively. The simulation time is set as 30 s, and the loops $1 \sim 5$ have the same physical parameters as System 1 & 3, and those of the loops $6 \sim 10$ are the same as System 2 & 4 in Tab. 1. Each time slot is 4.1 ms, and one superframe has 11 slots (one beacon frame). The control periods are $\mathcal{T} = [41.6\ 166.6\ 41.6\ 166.6\ 83.3\ 41.6\ 83.3\ 41.6\ 83.3\ 166.6]$ ms, and the numbers of time slots occupied by one transmission are $\mathcal{H} = [3\ 1\ 3\ 1\ 2\ 2\ 1\ 2\ 1\ 1]$. The rest of the experimental settings are the same as Sec. 8.1. The MAE is shown in Fig. 22, and the response curve, and sending & receiving messages of loop 6 by ADPS-H-3 are presented in Fig. 23.

Compared with the RR scheduling, Fig. 22 shows that ADPS-H-m can effectively improve the control performance in more complicated scenarios and large-scale dynamic network situations. ADPS-H and multi-horizon ADPS-H-m ($m = 2, 3, 4, 5$) improve performance by 16.5% and 20.8%~

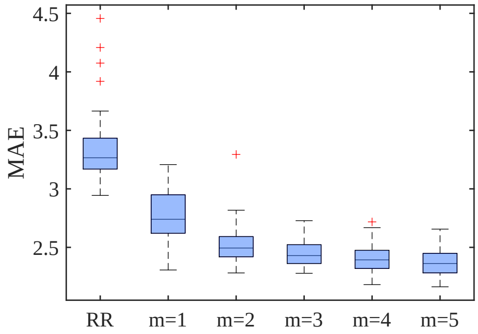
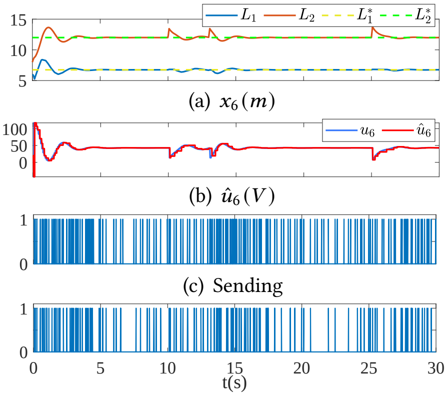


Fig. 22. MAEs of ADPS-H-m with different m under variable background noise levels (α=0.3).



(a) $x_6(m)$

(b) $\hat{u}_6(V)$

(c) Sending

(d) Receiving

Fig. 23. Response curve and network scheduling of loop 6.

27.8% with RR scheduling, respectively. Similar to the results presented in Figs. 18 and 19, when the physical disturbances happen at time 10, 13, and 25 s shown in Fig. 23(a), ADPS-H-m allocates more scheduling resources to loop 6 so that loop 6 can update the control commands on the actuation side in time in Fig. 23(b). While loop 6 is within the steady state, e.g., 5 ∼ 10 s and 17 ∼ 25 s in Fig. 23(a), ADPS-H-m would reallocate these resources to loops that are in the transition states by minimizing the overall control cost Eq. (21).

9 CONCLUSION

In this paper, we present comprehensive scheduling strategies for wireless control systems based on adaptive dynamic programming. Adaptive dynamic programming scheduling (ADPS) can effectively solve the optimal scheduling problem based on one-step control performance at runtime. Moreover, by incorporating a longer-horizon evolution of the physical plants, we extended ADPS to ADPS-m, which optimizes multi-step performance, in order to further improve the overall control performance. For heterogeneous data flows with different data rates and sizes, ADPS-H and ADPS-H-m are proposed based on ADPS and ADPS-m, respectively. All the optimal scheduling strategies can guarantee the mean-square stability of the overall system under mild assumptions. Finally, a systematic evaluation is performed based on four double water-tank systems and simulations of a realistic IEEE 802.15.4 wireless network with TOSSIM and MATLAB/Simulink. The simulation results demonstrate the advantages of these comprehensive optimal scheduling strategies in effectively improving the overall control performance at low computing costs.

A APPENDICES

A.1 Proof of Lemma 6.1 (the Optimality of ADPS-H-m)

Considering the schedule $S(t+1)$ derived by ADPS-H-m and the optimal schedule $S^*(t+1)$ which has the minimal $\mathbb{E}(J_{\boldsymbol{H},\boldsymbol{m}}(x(t)))$, we prove the optimality by contradiction. According to Eq. (22), the difference in control cost between $S(t+1)$ and $S^*(t+1)$ can be expressed as follows,

$$\mathbb{E}\Big(J_{\boldsymbol{H},\boldsymbol{m}}(x(t))\Big)\Big|_{S(t+1)} - \mathbb{E}\Big(J_{\boldsymbol{H},\boldsymbol{m}}(x(t))\Big)\Big|_{S^*(t+1)}$$
$$= \sum_{i_1}^{N_1} \sum_{j=n^*_{i_1}(t+1)+1}^{n_{i_1}(t+1)} \Big(\sum_{r=1}^{m_{i_1}} \sum_{g=0}^{2^r-1} \lambda_{g,r,i} \big(f(\mathcal{B}_{i_1}, g, r)^j - f(\mathcal{B}_{i_1}, g, r)^{j-1} \big) \Big)$$

$$-\sum_{i_2}^{N_2} \sum_{j=n_{i_2}(t+1)+1}^{n^*_{i_2}(t+1)} \Big(\sum_{r=1}^{m_{i_2}} \sum_{g=0}^{2^r-1} \lambda_{g,r,i} \big(f(\mathcal{B}_{i_2},g,r)^j - f(\mathcal{B}_{i_2},g,r)^{j-1} \big) \Big) \geq 0 \tag{39}$$

where

$$\begin{cases} n_{i_1}(t+1) - n^*_{i_1}(t+1) > 0 & i_1 \in \{1,2,...,N_1\} \\ n_{i_2}(t+1) - n^*_{i_2}(t+1) < 0 & i_2 \in \{1,2,...,N_2\} \\ n_{i_3}(t+1) - n^*_{i_3}(t+1) = 0 & i_3 \in \{1,2,...,N_3\} \\ N_1 + N_2 + N_3 = N'(t+1). \end{cases}$$

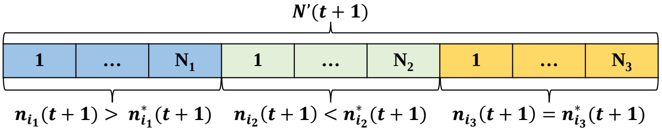

$N'(t+1)$
1
...
N_1
1
...
N_2
1
...
N_3
$n_{i_1}(t+1) > n^*_{i_1}(t+1)$
$n_{i_2}(t+1) < n^*_{i_2}(t+1)$
$n_{i_3}(t+1) = n^*_{i_3}(t+1)$

Fig. 24. The sets of each loop

Here $n^*_i(t+1)$ is the slots assigned to loop i, which is also the number of loop i occurrences in $S^*(t+1)$, $N'(t+1)$ is the number of loops which will be controlled at time $t+1$. The control loops are divided into three sets based on their numbers of allocated time slots, as shown in Fig. 24. Three groups indicate allocated time slots in $S(t+1)$ are more, less, or equal to that of the optimal scheduling result $S^*(t+1)$, respectively. Considering that loops are independent of each other, such a rearrangement does not affect the results.

Noting that $0 \leq f(\mathcal{B}_i,g,r) = \prod_{j=0}^{r-1} \mathcal{B}_i(t+1+j)^{\lfloor \frac{g}{2^j} \rfloor \mod 2} \leq 1$. $\forall i \in \{1,2,...,N'(t+1)\}, r \in \{1,2,...,m_i\}, g \in \{0,1,...,2^r-1\}$, we have

$$\lambda_{g,r,i}\big(f(\mathcal{B}_i,g,r)^{j+1} - f(\mathcal{B}_i,g,r)^j\big) \geq \lambda_{g,r,i}\big(f(\mathcal{B}_i,g,r)^j - f(\mathcal{B}_i,g,r)^{j-1}\big), \tag{40}$$

According to (39) and (40), we can get the inequlity (41).

$$\begin{aligned} &\mathbb{E}\Big(J_{\boldsymbol{H},\boldsymbol{m}}(x(t))\Big)\Big|_{S(t+1)} - \mathbb{E}\Big(J_{\boldsymbol{H},\boldsymbol{m}}(x(t))\Big)\Big|_{S^*(t+1)} \\ &\leq \sum_{i_1}^{N_1} \sum_{j=n^*_{i_1}(t+1)+1}^{n_{i_1}(t+1)} \Big(\sum_{r=1}^{m_{i_1}} \sum_{g=0}^{2^r-1} \lambda_{g,r,i}\big(f(\mathcal{B}_{i_1},g,r)^{n_{i_1}(t+1)+1} - f(\mathcal{B}_{i_1},g,r)^{n_{i_1}(t+1)}\big)\Big) - \\ &\quad \sum_{i_2}^{N_2} \sum_{j=n_{i_2}(t+1)+1}^{n^*_{i_2}(t+1)} \Big(\sum_{r=1}^{m_{i_2}} \sum_{g=0}^{2^r-1} \lambda_{g,r,i}\big(f(\mathcal{B}_{i_2},g,r)^{n_{i_2}(t+1)+1} - f(\mathcal{B}_{i_2},g,r)^{n_{i_2}(t+1)}\big)\Big) \\ &= \sum_{i_1}^{N_1} \big(n_{i_1}(t+1) - n^*_{i_1}(t+1)\big)\Big(\sum_{r=1}^{m_{i_1}} \sum_{g=0}^{2^r-1} \lambda_{g,r,i}\big(f(\mathcal{B}_{i_1},g,r)^{n_{i_1}(t+1)+1} - f(\mathcal{B}_{i_1},g,r)^{n_{i_1}(t+1)}\big)\Big) - \\ &\quad \sum_{i_2}^{N_2} \big(n^*_{i_2}(t+1) - n_{i_2}(t+1)\big)\Big(\sum_{r=1}^{m_{i_2}} \sum_{g=0}^{2^r-1} \lambda_{g,r,i}\big(f(\mathcal{B}_{i_2},g,r)^{n_{i_2}(t+1)+1} - f(\mathcal{B}_{i_2},g,r)^{n_{i_2}(t+1)}\big)\Big) \end{aligned} \tag{41}$$

As the ADPS-H-m algorithm and the optimal scheduling allocate the same number of time slots, then for any time t,

$$\sum_{i_1=1}^{N_1} \bar{H}_{i_1}\big(n_{i_1}(t) - n^*_{i_1}(t)\big) = \sum_{i_2=1}^{N_2} \bar{H}_{i_2}\big(n^*_{i_2}(t) - n_{i_2}(t)\big). \tag{42}$$

According to Eq. (42), (41) can be rewritten as (43).

The polynomials $P_{i_1}(n_{i_1})$ and $P_{i_2}(n_{i_2})$ in (43) can be regarded as the average unit control benefit of the time slots that be differently allocated between $S(t+1)$ and $S^*(t+1)$ according to Eq. (23). Assuming that $P_{i_1}(n_{i_1}) > P_{i_2}(n_{i_2})$, $\forall i_1 \in \{1,2,...,N_1\}, i_2 \in \{1,2,...,N_2\}$, which means

$$\begin{aligned}
&\mathbb{E}\Big(J_{\boldsymbol{H},\boldsymbol{m}}\big(x(t)\big)\Big)\Big|_{S(t+1)} - \mathbb{E}\Big(J_{\boldsymbol{H},\boldsymbol{m}}\big(x(t)\big)\Big)\Big|_{S^*(t+1)} \\
&\leq \sum_{i_1}^{N_1} \bar{H}_{i_1}\big(n_{i_1}(t+1) - n^*_{i_1}(t+1)\big)\Big(\frac{\sum_{r=1}^{m_{i_1}} \sum_{g=0}^{2^r-1} \lambda_{g,r,i}\big(f(\mathcal{B}_{i_1}, g, r)^{n_{i_1}(t+1)+1} - f(\mathcal{B}_{i_1}, g, r)^{n_{i_1}(t+1)}\big)}{\bar{H}_{i_1}}\Big) - \\
&\quad \sum_{i_2}^{N_2} \bar{H}_{i_2}\big(n^*_{i_2}(t+1) - n_{i_2}(t+1)\big)\Big(\frac{\sum_{r=1}^{m_{i_2}} \sum_{g=0}^{2^r-1} \lambda_{g,r,i}\big(f(\mathcal{B}_{i_2}, g, r)^{n_{i_2}(t+1)+1} - f(\mathcal{B}_{i_2}, g, r)^{n_{i_2}(t+1)}\big)}{\bar{H}_{i_2}}\Big) \\
&\triangleq \sum_{i_1}^{N_1} \bar{H}_{i_1}\big(n_{i_1}(t+1) - n^*_{i_1}(t+1)\big)\big(P_{i_1}(n_{i_1})\big) - \sum_{i_2}^{N_2} \bar{H}_{i_2}\big(n_{i_2}(t+1) - n^*_{i_2}(t+1)\big)\big(P_{i_2}(n_{i_2})\big)
\end{aligned} \tag{43}$$

$\sum_{i_1=1}^{N_1} \bar{H}_{i_1}(n_{i_1}(t+1) - n^*_{i_1}(t+1))$ slots which are allocated to the loop set $\{1, 2, ..., N_1\}$ by ADPS-H-m would result in a smaller value of control cost if they are allocated to the loop set $\{1, 2, ..., N_2\}$ as the optimal $S^*(t+1)$ schedules. However, as the ADPS-H-m shown in Alg. 3 minimizes the value of control cost for the entire superframe by Eq. (24), the ADPS-H-m would choose to schedule those slot to the loops in $\{1, 2, ..., N_2\}$ as the optimal scheduling $S^*(t+1)$ combining this full used slots. This contradiction makes $P_{i_1}(n_{i_1}) \leq P_{i_2}(n_{i_2})$ inevitable. Finally, combining Eq. (42) and $P_{i_1}(n_{i_1}) \leq P_{i_2}(n_{i_2})$, we can conclude that $\mathbb{E}(J_{\boldsymbol{H},\boldsymbol{m}}(x(t)))|_{S(t+1)} - \mathbb{E}(J_{\boldsymbol{H},\boldsymbol{m}}(x(t)))|_{S^*(t+1)} \leq 0$. As (39) shows that $\mathbb{E}(J_{\boldsymbol{H},\boldsymbol{m}}(x(t)))|_{S(t+1)} - \mathbb{E}(J_{\boldsymbol{H},\boldsymbol{m}}(x(t)))|_{S^*(t+1)} \geq 0$, thus $\mathbb{E}(J_{\boldsymbol{H},\boldsymbol{m}}(x(t)))|_{S(t+1)} - \mathbb{E}(J_{\boldsymbol{H},\boldsymbol{m}}(x(t)))|_{S^*(t+1)} = 0$, then $S(t+1) = S^*(t+1)$.

As the ADPS, ADPS-m, and ADPS-H are all special cases of the ADPS-H-m algorithm, the optimality of these algorithms under different system settings can also be guaranteed.

A.2 Proof of Lemma 7.3 (the Stability of ADPS-H-m)

In previous work [29], we have proven that Eq. (3) is a stochastic Lyapunov function of the optimal scheduling with one-step cost prediction and homogeneous data flows. Assuming that $x = \mathbf{0}$ is the equilibrium point of the system, when $x = \mathbf{0}$, $J_{\boldsymbol{H},\boldsymbol{m}}(\mathbf{0}) = \sum_{i=1}^{N}(J_{H_i,m_i}(0)) = 0$ is always satisfied. As each term in $J_{H_i,m_i}(x_i(t))$ is of the quadratic form of $x_i(t)$ and $\alpha \geq 0$, then $J_{H_i,m_i}(x_i(t)) > 0, \forall x \neq 0$. Also it is easy to see that when $\|x\| \to \infty$, $J_{\boldsymbol{H},\boldsymbol{m}}(x(t)) \to \infty$.

Then we need to prove that $\mathbb{E}(J_{\boldsymbol{H},\boldsymbol{m}}(x(t)))$ decreases along system trajectories, i.e., $\forall t, \mathbb{E}(J_{\boldsymbol{H},\boldsymbol{m}}(x(t+1))) \leq \mathbb{E}(J_{\boldsymbol{H},\boldsymbol{m}}(x(t)))$. The $\mathbb{E}(J_{\boldsymbol{H},\boldsymbol{m}}(x(t+1))) - \mathbb{E}(J_{\boldsymbol{H},\boldsymbol{m}}(x(t)))$ can be rewritten as Eq. (44).

$$\mathbb{E}\Big(J_{\boldsymbol{H},\boldsymbol{m}}\big(x(t+1)\big)\Big) - \mathbb{E}\Big(J_{\boldsymbol{H},\boldsymbol{m}}\big(x(t)\big)\Big) = \sum_{i=1}^{N}\sum_{r=1}^{m_i} \alpha^{r-1}\Big(\mathbb{E}\Big(J\big(x_i(t+r+1)\big)\Big) - \mathbb{E}\Big(J\big(x_i(t+r)\big)\Big)\Big) \tag{44}$$

As every single loop can check its MSS by the LMI in Lemma. 7.1, the quadratic control cost Eq. (3) can be a stochastic Lyapunov function [29], e.g., $J(x_i(t)) = x_i^T(t)P_i x_i(t)$, where P_i is derived by (30). Therefore, $\alpha^{r-1}\big(\mathbb{E}\big(J\big(x_i(t+r+1)\big)\big) - \mathbb{E}\big(J\big(x_i(t+r)\big)\big)\big) \leq 0$ is always satisfied for any i, r and $0 < \alpha \leq 1$. Hence the $\mathbb{E}(J_{\boldsymbol{H},\boldsymbol{m}}(x(t)))$ decreases along system trajectories and $J_{\boldsymbol{H},\boldsymbol{m}}(x(t))$ is a stochastic Lyapunov function of the multi-loop control system.

ACKNOWLEDGMENTS

This work is supported in part by the NSF of China under Grants 62103268, 92167205, 62202287, 61933009, 62025305, and Shanghai Chenguang Program 21CGA11.

REFERENCES

[1] 2020. IEEE Standard for Low-Rate Wireless Networks. *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)* (2020), 1–800. https://doi.org/10.1109/IEEESTD.2020.9144691

[2] Anders Ahlén, Johan Akerberg, Markus Eriksson, Alf J Isaksson, Takuya Iwaki, Karl Henrik Johansson, Steffi Knorn, Thomas Lindh, and Henrik Sandberg. 2019. Toward wireless control in industrial process automation: A case study at a paper mill. *IEEE Control Syst. Mag.* 39, 5 (2019), 36–57.

[3] D. Antunes and W. P. M. H. Heemels. 2014. Rollout Event-Triggered Control: Beyond Periodic Control Performance. *IEEE Trans. Autom. Control* 59, 12 (2014), 3296–3311.

[4] Duarte J Antunes, WPMH Heemels, João Pedro Hespanha, and CJ Silvestre. 2012. Scheduling measurements and controls over networks—Part I: Rollout strategies for protocol design. In *2012 American Control Conference (ACC).* IEEE, 2036–2041.

[5] Duarte J Antunes, WPMH Heemels, João Pedro Hespanha, and CJ Silvestre. 2012. Scheduling measurements and controls over networks—Part II: Rollout strategies for simultaneous protocol and controller design. In *2012 American Control Conference (ACC).* IEEE, 2042–2047.

[6] José Araújo, Manuel Mazo, Adolfo Anta, Paulo Tabuada, and Karl H Johansson. 2013. System architectures, protocols and algorithms for aperiodic wireless control systems. *IEEE Trans. Ind. Informat.* 10, 1 (2013), 175–184.

[7] Dominik Baumann, Fabian Mager, Romain Jacob, Lothar Thiele, Marco Zimmerling, and Sebastian Trimpe. 2019. Fast feedback control over multi-hop wireless networks with mode changes and stability guarantees. *ACM Trans. Cyber-Phys. Syst.* 4, 2 (2019), 1–32.

[8] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. 1994. *Linear matrix inequalities in system and control theory.* SIAM.

[9] Xiaotian Dai and Alan Burns. 2020. Period adaptation of real-time control tasks with fixed-priority scheduling in cyber-physical systems. *J. Syst. Architect.* 103 (2020), 101691.

[10] Burak Demirel, Zhenhua Zou, Pablo Soldati, and Mikael Johansson. 2014. Modular Design of Jointly Optimal Controllers and Forwarding Policies for Wireless Control. *IEEE Trans. Autom. Control* 59, 12 (2014), 3252–3265.

[11] Abdelouahid Derhab, Fatma Ounini, and Badis Remli. 2012. MOB-TOSSIM: An Extension Framework for TOSSIM Simulator to Support Mobility in Wireless Sensor and Actuator Networks. In *2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems.* 300–305.

[12] Felix Dobslaw, Tingting Zhang, and Mikael Gidlund. 2014. End-to-end reliability-aware scheduling for wireless sensor networks. *IEEE Trans. Ind. Informat.* 12, 2 (2014), 758–767.

[13] Mark Eisen, Mohammad M Rashid, Konstantinos Gatsis, Dave Cavalcanti, Nageen Himayat, and Alejandro Ribeiro. 2019. Control aware radio resource allocation in low latency wireless control systems. *IEEE Internet Things J.* 6, 5 (2019), 7878–7890.

[14] Mark Eisen, Mohammad M Rashid, Alejandro Ribeiro, and Dave Cavalcanti. 2020. Scheduling low latency traffic for wireless control systems in 5G networks. In *ICC 2020-2020 IEEE International Conference on Communications (ICC).* IEEE, 1–6.

[15] Luca Greco, Antoine Chaillet, and Antonio Bicchi. 2012. Exploiting packet size in uncertain nonlinear networked control systems. *Automatica* 48, 11 (2012), 2801–2811.

[16] Dolvara Gunatilaka and Chenyang Lu. 2020. REACT: an Agile Control Plane for Industrial Wireless Sensor-Actuator Networks. In *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI).* IEEE, 53–65.

[17] Jianlin Guo and Philip Orlik. 2017. Self-transmission control in IoT over heterogeneous wireless networks. In *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN).* 898–903.

[18] Victor Wattin Håkansson, Naveen KD Venkategowda, Stefan Werner, and Pramod K Varshney. 2022. Optimal Scheduling of Multiple Spatiotemporally Dependent Observations for Remote Estimation Using Age of Information. *IEEE Internet Things J.* 9, 20 (2022), 20308–20321.

[19] Erik Henriksson, Daniel E Quevedo, Edwin GW Peters, Henrik Sandberg, and Karl Henrik Johansson. 2015. Multiple-loop self-triggered model predictive control for network scheduling and control. *IEEE Trans. Control Syst. Technol.* 23, 6 (2015), 2167–2181.

[20] Kang Huang, Wanchun Liu, Yonghui Li, Andrey Savkin, and Branka Vucetic. 2020. Wireless feedback control with variable packet length for industrial IoT. *IEEE Wirel. Commun. Lett.* 9, 9 (2020), 1586–1590.

[21] Igor Kadota, Abhishek Sinha, and Eytan Modiano. 2019. Scheduling Algorithms for Optimizing Age of Information in Wireless Networks With Throughput Constraints. *IEEE/ACM Trans. Netw.* 27, 4 (2019), 1359–1372.

[22] Dohwan Kim, Yuchang Won, Seunghyeon Kim, Yongsoon Eun, Kyung-Joon Park, and Karl H Johansson. 2019. Sampling rate optimization for IEEE 802.11 wireless control systems. In *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS).* 87–96.

[23] Matthias Köppe. 2012. On the Complexity of Nonlinear Mixed-Integer Optimization. In *Mixed Integer Nonlinear Programming.*

[24] HyungJune Lee, Alberto Cerpa, and Philip Levis. 2007. Improving wireless simulation through noise modeling. In *Proceedings of the 6th international conference on Information processing in sensor networks (IPSN).* 21–30.

[25] Larry Leemis. 2020. *The Product of N Mutually Independent Bernoulli Random Variables is Bernoulli.* http://www.math.wm.edu/~leemis/chart/UDR/PDFs/BernoulliP.pdf

[26] Bo Li, Lanshun Nie, Chengjie Wu, Humberto Gonzalez, and Chenyang Lu. 2015. Incorporating emergency alarms in reliable wireless process control. In *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems (ICCPS)*. 218–227.

[27] Merid Lješnjanin, Daniel E Quevedo, and Dragan Nešić. 2014. Packetized MPC with dynamic scheduling constraints and bounded packet dropouts. *Automatica* 50, 3 (2014), 784–797.

[28] Chenyang Lu, Abusayeed Saifullah, Bo Li, Mo Sha, Humberto Gonzalez, Dolvara Gunatilaka, Chengjie Wu, Lanshun Nie, and Yixin Chen. 2015. Real-time wireless sensor-actuator networks for industrial cyber-physical systems. *Proc. IEEE* 104, 5 (2015), 1013–1024.

[29] Yehan Ma, Jianlin Guo, Yebin Wang, Ankush Chakrabarty, Heejin Ahn, Philip Orlik, Xinping Guan, and Chenyang Lu. 2022. Optimal Dynamic Transmission Scheduling for Wireless Networked Control Systems. *IEEE Trans. Control Syst. Technol.* 30, 6 (2022), 2360–2376.

[30] Fabian Mager, Dominik Baumann, Carsten Herrmann, Sebastian Trimpe, and Marco Zimmerling. 2022. Scaling beyond bandwidth limitations: Wireless control with stability guarantees under overload. *ACM Transactions on Cyber-Physical Systems (TCPS)* 6, 3 (2022), 1–30.

[31] Mohammad H Mamduhi, Dipankar Maity, Sandra Hirche, John S Baras, and Karl H Johansson. 2021. Delay-sensitive joint optimal control and resource management in multiloop networked control systems. *IEEE Trans. Control Netw. Syst.* 8, 3 (2021), 1093–1106.

[32] Prabhat K Mishra, Debasish Chatterjee, and Daniel E Quevedo. 2018. Sparse and constrained stochastic predictive control for networked systems. *Automatica* 87 (2018), 40–51.

[33] Prabhat K Mishra, Debasish Chatterjee, and Daniel E Quevedo. 2020. Stochastic predictive control under intermittent observations and unreliable actions. *Automatica* 118 (2020), 109012.

[34] Venkata Prashant Modekurthy and Abusayeed Saifullah. 2019. Online period selection for wireless control systems. In *2019 IEEE International Conference on Industrial Internet (ICII)*. IEEE, 170–179.

[35] Minjae Park, Sohyang Park, Thanh-Hai Nguyen, Dinh-Lam Pham, Dongkeun Oh, Kyoung-Sook Kim, In-Kyoo Chun, and Kwanghoon Pio Kim. 2022. A Time-Series Process Event Log Preprocessing Approach for Data-Intensive and Predictive Operationalization of Smart Factories. In *2022 24th International Conference on Advanced Communication Technology (ICACT)*. IEEE, 252–255.

[36] Edwin GW Peters, Daniel E Quevedo, and Minyue Fu. 2016. Controller and scheduler codesign for feedback control over IEEE 802.15. 4 networks. *IEEE Trans. Control Syst. Technol.* 24, 6 (2016).

[37] Debayan Roy, Sumana Ghosh, Qi Zhu, Marco Caccamo, and Samarjit Chakraborty. 2020. GoodSpread: Criticality-aware static scheduling of CPS with multi-QoS resources. In *2020 IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 178–190.

[38] Abusayeed Saifullah, Chengjie Wu, Paras Babu Tiwari, You Xu, Yong Fu, Chenyang Lu, and Yixin Chen. 2014. Near optimal rate selection for wireless control systems. *ACM Trans. Embed. Comput. Syst.* 13, 4s (2014), 1–25.

[39] Junyang Shi, Mo Sha, and Zhicheng Yang. 2019. Distributed Graph Routing and Scheduling for Industrial Wireless Sensor-Actuator Networks. *IEEE/ACM Trans. Netw.* 27, 4 (2019), 1669–1682.

[40] Jianping Song, Song Han, Al Mok, Deji Chen, Mike Lucas, Mark Nixon, and Wally Pratt. 2008. WirelessHART: Applying wireless technology in real-time industrial process control. In *2008 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 377–386.

[41] Rajat Talak, Sertac Karaman, and Eytan Modiano. 2020. Optimizing Information Freshness in Wireless Networks Under General Interference Constraints. *IEEE/ACM Trans. Netw.* 28, 1 (2020), 15–28.

[42] James W. Taylor. 2003. Exponential smoothing with a damped multiplicative trend. *Int. J. Forecast.* 19, 4 (2003), 715–725.

[43] Heng Wang, Xin Xie, Xiaozhe Li, and Jingqi Yang. 2022. Scheduling schemes for age optimization in IoT systems with limited retransmission times. *IEEE Internet Things J.* 9, 21 (2022), 21458–21468.

[44] Shuang Wu, Xiaoqiang Ren, Subhrakanti Dey, and Ling Shi. 2017. Optimal Scheduling of Multiple Sensors with Packet Length Constraint. *IFAC-PapersOnLine* 50, 1 (2017), 14430–14435.

[45] Bo Zhou and Walid Saad. 2019. Joint Status Sampling and Updating for Minimizing Age of Information in the Internet of Things. *IEEE Trans. Commun.* 67, 11 (2019), 7468–7482.

[46] Bo Zhou and Walid Saad. 2019. Minimum age of information in the Internet of Things with non-uniform status packet sizes. *IEEE Trans. Wirel. Commun.* 19, 3 (2019), 1933–1947.

[47] Marco Zimmerling, Luca Mottola, Pratyush Kumar, Federico Ferrari, and Lothar Thiele. 2017. Adaptive real-time communication for wireless cyber-physical systems. *ACM Trans. Cyber-Phys. Syst.* 1, 2 (2017), 1–29.

Received 18 August 2023; revised 02 January 2024; accepted 13 July 2024