

## Article

# Scheduling Strategy Design Framework for Cyber–Physical System with Non-Negligible Propagation Delay

Zuoyu An , Shaohua Wu \*, Tiange Liu , Jian Jiao  and Qinyu Zhang

Communication Engineering Research Centre, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China; 19s052043@stu.hit.edu.cn (Z.A.); 19s152063@stu.hit.edu.cn (T.L.); jiaojian@hit.edu.cn (J.J.); zqy@hit.edu.cn (Q.Z.)

\* Correspondence: hitwush@hit.edu.cn

**Abstract:** Cyber–physical systems (CPS) have been widely employed as wireless control networks. There is a special type of CPS which is developed from the wireless networked control systems (WNCS). They usually include two communication links: Uplink transmission and downlink transmission. Those two links form a closed-loop. When such CPS are deployed for time-sensitive applications such as remote control, the uplink and downlink propagation delay are non-negligible. However, existing studies on CPS/WNCS usually ignore the propagation delay of the uplink and downlink channels. In order to achieve the best balance between uplink and downlink transmissions under such circumstances, we propose a heuristic framework to obtain the optimal scheduling strategy that can minimize the long-term average control cost. We model the optimization problem as a Markov decision process (MDP), and then give the sufficient conditions for the existence of the optimal scheduling strategy. We propose the semi-predictive framework to eliminate the impact of the coupling characteristic between the uplink and downlink data packets. Then we obtain the lookup table-based optimal offline strategy and the neural network-based suboptimal online strategy. Numerical simulation shows that the scheduling strategies obtained by this framework can bring significant performance improvements over the existing strategies.

**Keywords:** cyber–physical system; wireless networked control system; remote control; communication control co-design; age of information



**Citation:** An, Z.; Wu, S.; Liu, T.; Jiao, J.; Zhang, Q. Scheduling Strategy Design Framework for Cyber–Physical System with Non-Negligible Propagation Delay. *Entropy* **2021**, *23*, 714. <https://doi.org/10.3390/e23060714>

Academic Editors: Anthony Ephremides and Yin Sun

Received: 6 May 2021

Accepted: 2 June 2021

Published: 4 June 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In the recent past, applications of the wireless control networks have become more and more extensive, such as drone formations, autonomous vehicles, automatic factories, etc. Some of those scenarios implicate new requirements for remote control technology, which is a sub-topic of communication control co-design. Remote control technology originates from wireless control systems with long propagation delay such as far-sea monitoring and high-efficiency satellite IoT. The main cause of long propagation delay is the large-scale geographic distance. This feature makes it extremely challenging to design CPS under this scenario. In order to meet the need of remote control with propagation delay, that is, to maintain stable closed-loop control and reduce control costs, we propose a new framework to design uplink and downlink scheduling strategies.

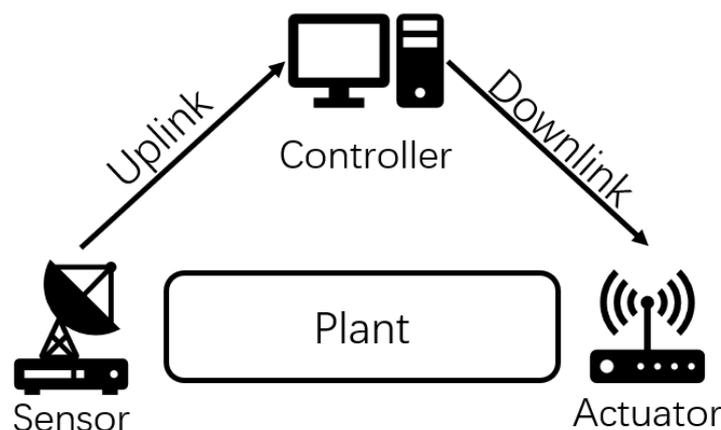
As show in Figure 1, a typical CPS deployed under the single closed-loop control scenario contains a control system and a communication system. In the rest of this article, we use single-loop CPS to refer to this specific type of CPS. The communication process of a typical single-loop CPS can be divided into two parts: Uplink sensor transmission and downlink controller transmission. The uplink transmission is initiated by the sensor and sends the state update packet from the plant to the controller. The controller first uses this data to obtain a more accurate estimate of the factory status. Then the downlink transmission is initiated to send command information from the controller to the actuator located at the factory. The actuator acts on the factory to maintain the factory's stability.

Taking into account the characteristics of a control system, the command can only be generated with an accurate estimation, which means the downlink transmission must occur after a successful uplink transmission. Because of this fixed timing relationship, CPS has to work in half-duplex in most cases: namely, only one of the uplink sensor transmission and the downlink controller transmission can be activated to send a data packet in the same time slot. That means there is a problem of how to design a scheduling strategy between those two transmissions. Note that the uplink and downlink channels here are not just a single wireless channel, but a simplified modeling of a fixed routing link with multiple relays. This scenario is for some special remote control systems that use satellites as relays. Therefore, the propagation delay in our paper is essentially a collection of various delays contained in the entire relay link, including processing delay, transmission delay, propagation delay, etc. This unified modeling is used because the link characteristics of a fixed routing multi-relay link can be described by an equivalent link with a specific code error rate and propagation delay.

There are many related works about WNCS and CPS [1–4]. Focusing on the conflict of the accuracy requirements of control systems and the limited quantization level [5], proposed the application of dynamic quantization technology in the communication control co-design. Some works designed CPS with the limitation of wireless coding process, such as code length allocation [6,7], code length design [8,9] and adaptive code length adjustment [10]. Considering the fading characteristics of transmission channels, studies of adaptive transmit power adjustment technology by predicting the fast or slow fading of transmission channels are proposed in [11,12]. Some of the above studies include the idea of designing CPS for time-sensitive applications. Nowadays, the most widely used measure of timeliness is Age of Information (AoI) [13], which is defined as the time elapsed since a certain data packet was generated:

$$\Delta(t) = t - t' \quad (1)$$

where  $t$  represents the current time,  $t'$  represents the time when the packet was generated. It used to be very difficult to express the control performance measurement, that is, the system state mean square error (MSE) [14] when the control system and the communication system are combined. The proposal of AoI changed this situation. For example, the system state MSE of a linear time invariant system (LTI) can be simply expressed as a function of AoI. This improvement greatly reduces the difficulty of describing the overall system performance in the communication control co-design scenario [15,16].



**Figure 1.** Cyber-physical system deployed under the single closed-loop control scenario.

Based on AoI, many related studies have been derived, such as the application of the HARQ mechanism for single-loop CPS to improve the overall timeliness [17,18], and the scheduling strategy aiming to minimize the long-term average MSE for single-loop

CPS without transmission delay [19]. Some studies about the multi-loop scheduling strategy design aiming at optimizing timeliness have also been proposed. Reference [20] focuses on the design of the data inter-arrival rate and code length allocation strategy. References [21,22] proposed the uplink scheduling strategy of multi-loop WNCS under the ideal assumption of downlink transmission. Furthermore, the authors of [23,24] discuss the application of data packet transmission result prediction technology in WNCS design.

The scenarios studied above concern mainly short-distance Industrial Internet of Things (IIoT), so the impact of uplink and downlink propagation delay on the closed-loop control performance of a CPS is generally ignored. Besides, the above studies only consider one of the two code error rates of the uplink and the downlink transmission. Under the remote control scenario, the code error rates and propagation delay of both links are not only non-negligible, but also have a huge impact on the overall performance of the single-loop CPS. Some works have studied the design of WNCS optimal control strategy under time-delay scenarios [25–27]. However, they do not consider the impact of the code error rate and the scheduling strategy which are issues that cannot be ignored in the design of communication systems in the field of communication engineering. To this end, we propose a new framework to obtain the optimal scheduling strategy while considering both the code error rates and propagation delay. This strategy can minimize the long-term average control cost.

Firstly, we model the single-loop CPS as an MDP problem and give the sufficient conditions for the stability of CPS. Secondly, we propose a heuristic semi-predictive framework to eliminate the impact of the coupling characteristic between the uplink and downlink data packets. Finally, we obtain the lookup table-based optimal offline strategy and the neural network-based suboptimal online strategy for the single-loop CPS. The whole process can be expanded according to actual deployment requirements with any fixed propagation delay as long as the sufficient condition is satisfied.

The rest of this paper is organized as follows: In Section 2, we provide the system model and formulate the optimization problem. In Section 3, we introduce the semi-predictive framework and transform the optimization problem into an MDP problem. In Section 4, we obtain the optimal offline strategy and the suboptimal online strategy. In Section 5, we show the numerical simulation results. We conclude this work in Section 6.

## 2. System Model

### 2.1. The Plant of the Single-Loop CPS

First, we model the plant in the single-loop CPS as a discrete-time LTI system:

$$X_{k+1} = AX_k + BU_k + Z_k, \forall k \quad (2)$$

where  $k$  represents the  $k$ -th time slot,  $X_k \in \mathbb{R}$  represents the state of the plant at time slot  $k$ ,  $U_k \in \mathbb{R}$  represents the executed control command,  $Z_k \in \mathbb{R}$  represents the normally distributed plant noise whose mean and variance are  $\bar{z}$  and  $R$ , respectively.  $A \in \mathbb{R}$  represents the state transition coefficient,  $B \in \mathbb{R}$  represents the command control coefficient. We assume that the plant state remains unchanged within a single time slot. The goal of CPS is to maintain  $X$  around 0.

### 2.2. The Communication Process of the Single-Loop CPS

In the previous subsection, we explained that the entire single-loop CPS works in the half-duplex mode. Now we will explain the communication process of the single-loop CPS. The entire system adopts a centralized scheduling scheme because this scheme is more suitable for single-loop CPS. Under this scheme, the scheduling decision of uplink and downlink transmission is completely determined by the remote controller. We use  $a_k$  to represent the scheduling decision made by the controller in the time slot  $k$ . If the controller schedules uplink transmission in the slot  $k$ ,  $a_k = 1$ . If the controller schedules downlink transmission in the slot  $k$ ,  $a_k = 2$ . We assume that the code error rate of the uplink and downlink transmission channels are  $p_s, p_c \in (0, 1)$ , respectively. Both code error rates are

constant which means the uplink and downlink transmission fails with probability  $(p_s, p_c)$  in any time slot, respectively. Then we use  $\delta_k$  to represent the transmission result of the packet sent in the time slot  $k$ . No matter which transmission is scheduled, if it succeeds, then  $\delta_k = 1$ . Otherwise,  $\delta_k = 0$ . Since the processing procedures of most actual CPS are digital, the packets that have experienced a certain delay will start to be processed in the next processing cycle after it is received; we model the propagation delay of the uplink and downlink channel integer time slots  $d_{up}, d_{down} \in \mathbb{R}$ , respectively. To simplify the analysis, we assume that the transmission of scheduling instructions and feedback information is ideal.

In addition to the variables described above, we define the following two parameters to describe the status of each part in a single-loop CPS:

(1) State Estimation Age  $\tau_k$ : This is defined as the age of the latest valid uplink state update packet successfully received by the controller at the end of the time slot  $k$ .  $\tau_k$  reflects the accuracy of the estimation maintained by the remote controller. Because of the uplink propagation delay, the minimum value of state estimation age is  $d_{up}$ . When the specific time slot is not considered, it is abbreviated as  $\tau$ . Its update rule is as follows:

$$\tau_{k+1} = \begin{cases} d_{up} & \text{if } (a_j = 1) \& (\delta_j = 1) \\ \tau_k + 1 & \text{otherwise} \end{cases} \quad (3)$$

where  $j = k - d_{up} + 1$ .

(2) State Control Age  $\varphi_k$ : This is defined as the age of the uplink packet used to generate the latest successfully received downlink packet by the actuator at the end of the time slot  $k$ . This parameter represents the total time it takes for the entire CPS to complete a closed-loop control process. It reflects the degree of divergence of the plant's state. Because of the uplink and downlink propagation delay, the minimum value of the state control age is  $d_{up} + d_{down}$ . When the specific time slot is not considered, it is abbreviated as  $\varphi$ . Its update rule is as follows:

$$\varphi_{k+1} = \begin{cases} \tau_q + d_{down} & \text{if } (a_q = 2) \& (\delta_q = 1) \\ \varphi_k + 1 & \text{otherwise} \end{cases} \quad (4)$$

where  $q = k - d_{down} + 1$ . The abbreviations  $j$  and  $q$  will be used in the rest of this paper. Note that we set the initial values of  $\tau_0$  and  $\varphi_0$  to be 2. These values can be arbitrarily selected within a reasonable range. This is because the long-term average cost we focus on is not affected by those initial values.

### 2.3. The Control Process of the Single-Loop CPS

In this subsection, we will explain the control process of the single-loop CPS in detail, which is mainly completed by the remote controller and the actuator. The task of the remote controller can be divided into three parts: Maintaining state estimation, generating control commands, and scheduling uplink and downlink transmissions, while the actuator has only one task: Executing the received control commands.

(1) Maintaining State Estimation: We assume that the sensor can sample the state of the plant without distortion. The uplink transmission cannot be scheduled in every time slot. What is more, the scheduled transmission can fail because of the code error occurring during its propagation process. So the remote controller cannot receive a new state update packet in every time slot. Under these circumstances, the remote controller has to update the estimation  $\tilde{X}_k$  of the plant state  $X_k$  through the following process:

$$\tilde{X}_{k+1} = \begin{cases} g^{d_{up}}(X_j, k) & \text{if } (a_j = 1) \& (\delta_j = 1) \\ A\tilde{X}_k + BU_k & \text{otherwise} \end{cases} \quad (5)$$

where  $g(X, k) = AX + BU_k$ ,  $g^n(X, k) = g(g^{n-1}(X, k-1), k) \forall n > 1$ , and  $g^1(X, k) = g(X, k)$ . In this scenario, this estimation method has been proven to be optimal [28]. When a certain uplink transmission is successful, the remote controller can use the plant

state  $X_{k-d_{up}+1}$ , which is the exact value for  $d_{up} - 1$  time slots before, to obtain the state estimation  $\tilde{X}_{k+1}$  of the next time slot. When the current time slot has no successful uplink transmission, the controller can only update  $\tilde{X}_{k+1}$  with  $\tilde{X}_k$ . According to this process, we can derive the state estimation MSE of the remote controller as  $\tilde{Q}_k$ :

$$\tilde{Q}_k = \mathbb{E}[(\tilde{X}_k - X_k)^2] \tag{6}$$

Note that the state estimation error of the remote controller is entirely caused by the noise  $Z_k$ . By using the state estimation age  $\tau_k$ , we can rewrite the state estimation MSE as a recursive function of the noise variance  $R$ :

$$\tilde{Q}_{k+1} = \begin{cases} f(d_{up}) & \text{if } (a_j = 1) \& (\delta_j = 1) \\ f(\tau_k + 1) & \text{otherwise} \end{cases} \tag{7}$$

where  $f(x) = \sum_{i=1}^x (A^2)^{i-1} R$ . Equation (6) uses the definition of AoI to derive the MSE of the estimation. This representation greatly reduces the difficulty of calculation. In the following part, we will use the same idea to derive the single-loop CPS control performance metrics.

(2) Control Command Generation and Execution: In each time slot, while the remote controller maintains the state estimation, it also uses the estimation to generate a control command  $\tilde{U}_k$ :

$$\tilde{U}_k = K\tilde{X}_k \tag{8}$$

where  $K$  is the command generation coefficient. The goal of this control process is to maintain the state around 0. Since the downlink transmission has a propagation delay of  $d_{down}$  time slots, we must ensure  $BK = -A^{d_{down}}$ . To simplify the analysis, we set  $B = -A^{d_{down}}$ ,  $K = 1$ . Due to the code error rate and scheduling decisions, not every control command  $\tilde{U}$  can be received by the actuator. Only those scheduled and successfully transmitted can be used by the actuator. Therefore, the control command executed by the actuator is  $U_{k+1}$ :

$$U_{k+1} = \begin{cases} \tilde{U}_q & \text{if } (a_q = 2) \& (\delta_q = 1) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where  $q = k - d_{down} + 1$ . This control method shown by (8) and (9) is called single-step control, which is a common form in the field of classic cybernetics. Using this method, when a control command is successfully delivered to the actuator, the actual state value will return to a value as close to 0 as possible at one time. Such a process can maximize the effect of a single instruction.

(3) Single-Loop CPS Control Performance Metrics: Consistent with the estimation performance metrics, the control performance metrics is defined as the state MSE of the plant  $Q_k$ :

$$Q_k = \mathbb{E}[X_k^2] \tag{10}$$

Similar to  $\tilde{Q}_k$ , we can rewrite  $Q_k$  as a function of noise variance  $R$  and state control age  $\varphi$ :

$$Q_{k+1} = \begin{cases} f(\tau_q + d_{down}) & \text{if } (a_q = 2) \& (\delta_q = 1) \\ f(\varphi_k + 1) & \text{otherwise} \end{cases} \tag{11}$$

According to the control cost given by Equation (11), we can obtain the long-term average control cost, that is, the long-term average plant state MSE:

$$J = \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=0}^K Q_k \tag{12}$$

Equation (12) reflects the state deviation in the field of classic cybernetics which is the core cost metrics we care about. Please note that this parameter used to be very difficult to quantify without the introduction of AoI. Under certain conditions, the limit contained in Equation (12) may not exist, and the problem is unsolvable. In order to prevent such situations, the sufficient condition for the stability of WNCS with propagation delay will be given later, namely equation (19). In this paper, the scheduling strategy will be designed on the premise that equation (19) is satisfied.

(3) Uplink and Downlink Scheduling Process: In the previous subsection, we introduced the control performance measurement of a single-loop CPS. Now we will describe the scheduling process in detail. It has been explained that a single-loop CPS has two communication scenarios—the uplink transmission and the downlink transmission—and we can only choose one of them in each time slot under half-duplex mode. According to the previous definition, the scheduling decision of time slot  $k$  is recorded as  $a_k$ . The set of scheduling decisions of all time slots is called a scheduling strategy:

$$\pi \triangleq (a_1, a_2, \dots, a_k, \dots) \in \Pi \quad (13)$$

where  $\Pi$  represents the set of all scheduling strategies. Different scheduling strategies can significantly affect the control performance of a single-loop CPS. Every scheduling strategy  $\pi$  has its corresponding long-term average control cost  $J_\pi$ . Among all scheduling strategies, there is an optimal strategy  $\pi^* \in \Pi$ , which satisfies:

$$J_{\pi^*} \leq J_\pi, \forall \pi \in \Pi \quad (14)$$

Therefore, we can construct the following optimization problem. The goal of this problem is to minimize the long-term average plant state MSE to obtain the optimal scheduling strategy while taking transmission propagation delay and code error rates of two wireless channels into account, namely

$$\min_{\pi} \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=0}^K Q_k \quad (15)$$

### 3. Semi-Predictive Framework and MDP Modeling

In this section, we will introduce the coupling characteristic between the uplink and downlink data packets which is caused by their propagation delay. In the following paper, we will use the coupling characteristic to refer to the coupling characteristic between the uplink and downlink data packets to save space. We propose a semi-predictive framework to eliminate the effect of the coupling characteristics on the solution of optimization problem (15). Based on this framework, we remodel this optimization problem to an MDP problem. Note that the semi-predictive framework we proposed is suitable for any value of the uplink and downlink propagation delay. For the generality, we use  $d_{\text{up}} = d_{\text{down}} = 1$  as an example to illustrate the scheduling strategy design process. In the actual applications with different propagation delay, we only need to modify the value of  $d_{\text{up}}, d_{\text{down}}$  and adjust some parameters in the following modeling step to meet specific design requirements.

#### 3.1. The Packet Outdate Problem

Section 2 introduced the control mechanism of a single-loop CPS. Through the above analysis, it is easy to see that state update packets and control command packets have strong coupling characteristic for single-step control methods. Actually, such a coupling characteristic exists in any closed-loop control scenario as long as there exists propagation delay. This characteristic will cause some successfully delivered packets to become outdated. As shown in Figure 2, the green and red arrows represent state update packets  $\text{up}_1$  (left green arrow),  $\text{up}_2$  (left red arrow) and the control command packets  $\text{down}_1$  (right green arrow),  $\text{down}_2$  (right red arrow), respectively. The command  $\text{down}_1$  is generated by the controller using  $\text{up}_1$ , while  $\text{down}_2$  is generated by the controller using  $\text{up}_2$ . During the

period from the slot  $up_2$  sent to the slot  $down_2$  executed, if  $down_1$  is executed successfully, both  $up_2$  and  $down_2$  become invalid. In time slot 4,  $down_1$  is executed; the result is that the real state of the plant was returned to a value around 0. This process causes an interruption in the state estimation process which means the estimation updated by  $up_2$  is no longer accurate, so  $up_2$  is outdated. Since  $up_2$  is outdated, the control command  $down_2$  which was generated from it is also outdated. This is the main effect of the coupling characteristic and we named it the packet outdate problem.

As we can see, this problem is mainly caused by the discontinuity in the dynamic process of the plant. The discontinuity only occurs when a downlink control command is executed, which means the uplink state update packet will not cause this problem. When this happens, the outdated uplink and downlink data packets require different processing methods. For an outdated downlink packet, it only needs to be discarded. However, for an outdated uplink packet, we have to backtrack the state estimation before this outdated packet is used. We show the evolution of the state estimation age and state control age in Figure 2. It can be seen that the state estimation age has been backtracked by changing from  $\tau(3) = 2$  to  $\tau(4) = 4$ . The state control age will not be updated like this.

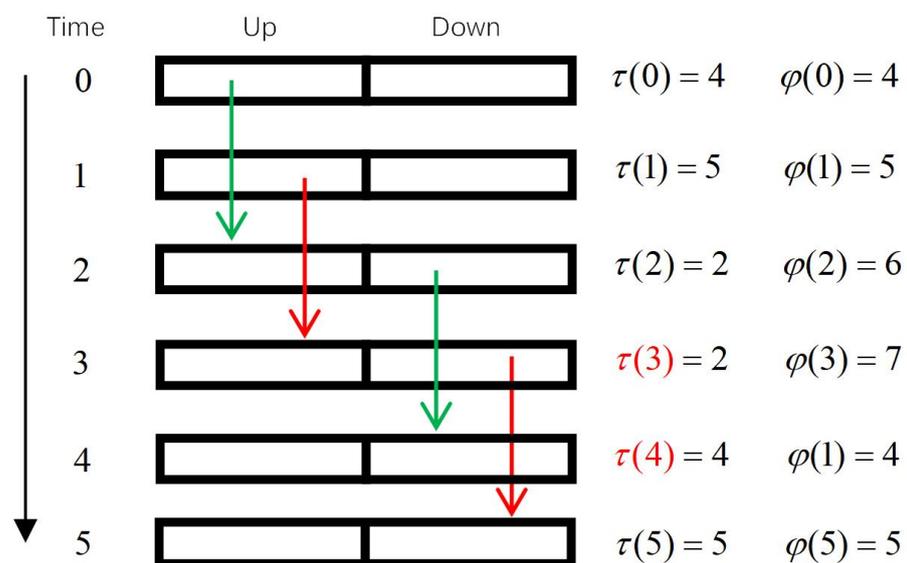


Figure 2. Analysis of Packet Outdated Phenomenon.

### 3.2. Main Idea of the Semi-Predictive Framework

In the previous subsection, we explained that the packet outdate problem has an impact on the update of the state estimation age, but this problem does not affect the update of the state control age. Therefore, when we try to construct a theoretical analysis framework, as long as the state control age is correct, the final analysis result can be guaranteed to be correct. In other words, the state estimation age of some time slots is allowed to deviate from the actual physical process. As long as it can be ensured that the state estimation age is accurate when the downlink data packet arrives at the actuator, the correct theoretical analysis can be guaranteed. It can be seen that it is possible to skip the state estimation age backtracking process in the theoretical analysis by using this feature. This is the main idea of the semi-predictive framework.

In the normal communication process, the decoding result of a data packet can only be determined after it arrives at the destination. For an uplink data packet, only after it arrives at the controller can it be known whether the data packet can be successfully decoded, while for a downlink packet, only after it arrives at the actuator can it be known whether the data packet can be successfully decoded. However, under the semi-predictive framework, we assume that the transmission result of a downlink packet is known as soon as the downlink packet is sent. Note that we do not predict the result of an uplink packet.

This is because the execution of the downlink command is the root cause of the packet outdated problem.

Take the case of Figure 2 as an example again; if we can foresee that the downlink control command packet  $\text{down}_1$  can be successfully decoded and is not outdated, then during the period from its sending to its arrival, any packets sent or arrived can be directly discarded since they will be outdated by  $\text{down}_1$ . Through this process, the impact of the packet outdated problem is eliminated and state estimation age backtracking is avoided.

While the update process of the state estimation age under the semi-predictive framework is different from the actual physical process, the scheduling strategy obtained based on this framework can still be directly applied to an actual physical process. In the actual physical process, if a downlink data packet arrives at the actuator successfully and is not outdated, then the uplink and downlink transmissions scheduled during its transmission must be outdated. In other words, no matter what scheduling decision the controller made, those packets sent during this period will be outdated. In other words, those scheduling decisions can be arbitrary since they do not affect the final result. Assuming that the downlink control command packet  $\text{down}_1$  in Figure 2 can be successfully decoded and not outdated, we will explain both age update processes under the semi-predictive framework and the actual physical process in detail.

(1) Semi-Predictive Framework: If  $\text{down}_1$  can be successfully decoded and not outdated, then the controller knows that it does not matter whether it chooses uplink or downlink during the transmission of  $\text{down}_1$  because those scheduled packets will be outdated anyway. Under these circumstance, a reasonable scheduling strategy is to regularly schedule one of the uplink and downlink transmissions during this period to consume time.

(2) Non-Predictive Framework (Actual Physical Process): In the actual physical process, during the transmission of  $\text{down}_1$ , the controller continues to schedule uplink or downlink transmissions according to a certain strategy. However, when  $\text{down}_1$  is received and decoded successfully, the previous scheduled transmissions of the controller are all outdated. So in the end, the scheduled transmissions during this period only consume time and have no practical effect.

It can be observed that, under the semi-predictive framework and the actual non-predictive scheduling, the single-loop CPS transmission results are uniform; that is, it is accurate to use the semi-predictive framework in the theoretical design and directly apply the results to the real applications. This subsection qualitatively analyzes the unity of the semi-predictive framework and the actual physical process. In the next subsection, we will quantitatively illustrate how this framework corresponds to actual physical processes through MDP modeling.

### 3.3. MDP Modeling of the Semi-Predictive Framework

Based on the semi-predictive framework, we model the single-loop CPS with uplink and downlink propagation delay as an MDP process with the following four elements:

(1) State Space: The state space of this MDP is

$$\mathbb{S} \triangleq \{a'(-d_{\max} + 1), \dots, a'(-1), a'(0), D(0), \tau(0), \varphi(0)\} \quad (16)$$

where  $d_{\max} = \max\{d_{\text{up}}, d_{\text{down}}\}$ ,  $D(n) \in \{0, 1, \dots, d_{\text{down}} + 1\}$ .  $a(n)$  represents the scheduling decision made in the time slot  $n$ .  $D(n)$  represents the time interval between the time slot when the latest valid downlink command packet (successfully transmitted and not outdated) in the time slot  $n$  was generated and the current time slot  $n$ .  $\tau(n)$  and  $\varphi(n)$  represent the state estimation age and the state control age at the time slot  $n$ , respectively. The time slot  $n$  is based on the current time slot: The time slot for which scheduling decisions are being made. Taking  $a'(-1)$  as an example: It represents the transmission action taken in the previous time slot of the current time slot. We set both the uplink and downlink propagation delay to be 1 for illustration in the rest of this paper, so the corresponding state space is:  $\mathbb{S} \triangleq \{a'(0), D(0), \tau(0), \varphi(0)\}$ . In the subsequent sections of this paper, the state space is abbreviated as  $\mathbb{S} \triangleq \{a', D, \tau, \varphi\}$  to save space.

(2) Action Space: The action space is  $\mathbb{A} \triangleq \{0, 1\}$ . This action space corresponds to the scheduling action  $a_k$ . If the controller schedules uplink transmission in the slot  $k$ ,  $a_k = 1$ . If the controller schedules downlink transmission in the slot  $k$ ,  $a_k = 2$ .

(3) State Transition Probability Matrix: The transition matrix is  $P(s'|s, a)$ . The state transition probability is the probability that the next state is  $s'$  by taking action  $a$  in the current state  $s$ . The transition probability is determined by the channel code error rate. According to the different parameter pairs:  $(a', D)$  in the state  $\mathbb{S}$ , the state transition matrix can be divided into five parts:  $(a', D) = [(1, 1), (1, 2), (2, 0), (2, 1), (2, 2)]$ . The complete construction rules are given in Appendix A.

(4) Cost Function: It can be seen from (4) and (11) that the cost function in a specific state is independent of the action. The cost function can be expressed as a function of the state control age  $\varphi_k$ :

$$C(s, a) = Q_k(s) = f(\varphi_k) \quad (17)$$

In the MDP modeling of the semi-predictive framework, the core parameter is  $D(n)$ . We limit its maximum value to  $d_{\text{down}} + 1$  because we only need to track the downlink transmissions in the past  $d_{\text{down}}$  time slots to ensure that we do not miss any possible packet outdated problems. Besides, such process can help to reduce the scale of the state space. The update rule of  $D(n)$  is as follows:

$$D_{k+1} = \begin{cases} 0 & \text{if } (a_k = 2) \& (\delta_k = 1) \\ \max(d_{\text{down}} + 1, D_k + 1) & \text{otherwise} \end{cases} \quad (18)$$

This updated process reflects the main idea of the semi-predictive framework and guarantees that it will not cause any differences between the state control ages of the theoretical analysis and the actual physical processes. In the next section, we will use the semi-predictive framework to design the optimal scheduling strategy.

#### 4. Online and Offline Scheduling Strategies

In this section, we first give the sufficient condition for the existence of the optimal scheduling strategy. Then we use the relative value iteration algorithm to obtain the lookup table-based optimal offline strategy. Aiming at reducing the space complexity of the algorithm and saving space for storing the optimal offline strategy, we further propose a neural network-based suboptimal online strategy. For different uplink and downlink propagation delay, the acquisition process of both strategies is universal, which means that the semi-predictive framework has high practical application value.

##### 4.1. Sufficient Conditions for the Strategies' Existence

**Theorem 1.** (Sufficient conditions for the stability of multi-loop half-duplex CPS with fixed uplink and downlink propagation delay.) Assuming there are  $K$  single-loop CPS, all of them share the same controller and form up a multi-loop CPS. If the controller can only schedule  $L$  uplink transmissions or  $L$  downlink transmissions in each time slot, then for each single-loop CPS  $i$ , if the code error probability of its corresponding uplink and downlink channels satisfies

$$\max\{p_{i,up}, p_{i,down}\} < \left(\frac{1}{(A_i)^2}\right)^{\lceil K/L \rceil}, i \in \{1, 2, \dots, K\} \quad (19)$$

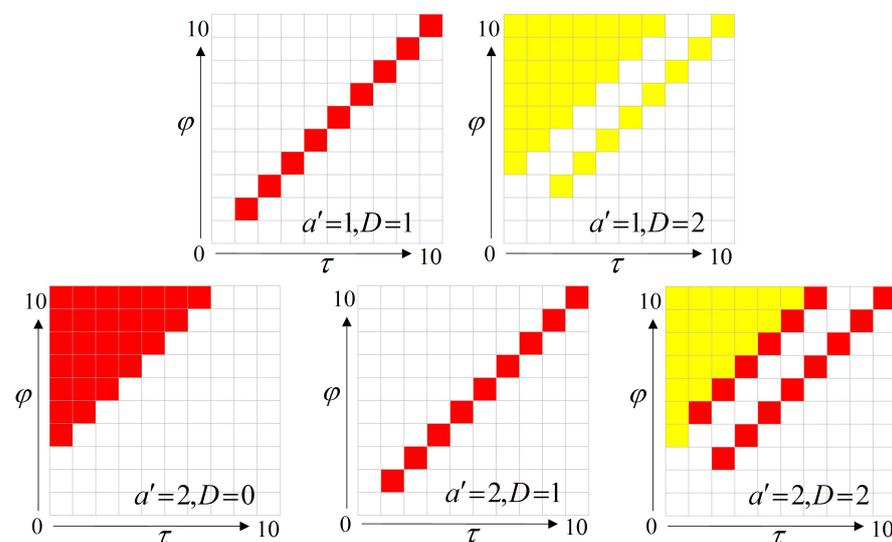
then there must exist a stationary deterministic scheduling strategy that can stabilize the multi-loop CPS. This stability remains as long as the uplink and downlink propagation delay are fixed, but the long term control performance metrics converge to a larger value with the increase of the propagation delay. When  $K = 1$ ,  $L = 1$  the above multi-loop CPS is just a single-loop CPS. The proof is given in Appendix B.

The essence of this sufficient condition is to link the instability of the control system with the reliability of the communication system. When the reliability of the communication

system is higher than the instability of the control system, an optimal scheduling strategy can be found for the communication system to meet the needs of the control system. This condition can effectively guide the design of single- and multi-loop CPS.

#### 4.2. Lookup Table-Based Optimal Offline Strategy

Since there is no theoretical upper limitation for the state estimation age and the state control age, the scale of the MDP state space is infinite, so it must be truncated before solving. We select  $N = \max\{\tau, \varphi\}$  as the truncation condition, and use the relative value iteration algorithm to solve the MDP problem. When the value of  $N$  is appropriate, this truncation will have no effect on the control performance. Such a suitable  $N$  can be obtained by conducting Monte Carlo experiments. In this section, we take  $N = 10$  as an example to show the resulting scheduling strategy in Figure 3.



**Figure 3.** Optimal Off-line Policy with  $N = 10$ . Red squares represent action  $a = 1$ ; yellow squares represent action  $a = 2$ .

In Figure 3, those red squares represent that the controller schedules uplink transmission in the corresponding state, and the yellow squares represent that the controller schedules downlink transmission in the corresponding state. As shown in Figure 3a,c,d, if  $D = \{0, 1\}$ , no matter which transmission is scheduled, the related packet will be outdated. So under this circumstance, the scheduling strategy can choose any action arbitrarily. Since we chose the relative value iterative algorithm to solve the MDP problem, the strategy we obtained chooses to use uplink transmission to fill these unnecessary transmissions. Note that this part corresponds to the description of Section 3 part C. We take  $\text{down}_1$  as an example again: In the actual physical process, it is not known that the next two transmissions are unnecessary transmissions after  $\text{down}_1$  is sent. The controller does not know that  $D = \{0, 1\}$ . Instead, it thinks that  $D$  is still equal to 2 at those time slots. Therefore, the controller continues to schedule according to the scheduling strategy. However,  $\text{down}_1$  makes those two packets outdated when it is executed, while for those states whose  $D = 2$ , the controller can make a scheduling decision with the right state information. The entire process makes sure that the actual process is consistent with the theoretical process.

After obtaining this scheduling strategy, it is stored as a lookup table by the controller and does not require any extra calculation ability from the controller, so we call it an offline strategy. However, since the iterative algorithm is a model-based algorithm, as  $N$  gradually increases, the scale of the state space  $N_S = 2 \cdot 3 \cdot N \cdot N = 6N^2$  in the MDP modeling increases exponentially. This leads to a sharp increase in the space complexity of the solving process and the lookup table could be too large to be stored. In order to

solve these problems, we propose an improved scheme based on neural network in the next subsection.

#### 4.3. Neural Network-Based Suboptimal Online Strategy

In Section 3, we remodeled the optimization problem to an MDP problem, and solved it to obtain the optimal offline strategy in the previous subsection. The optimal offline scheduling strategy based on the lookup table has two obvious shortcomings: The size of the lookup table increases linearly as the total number of states in the state space increases and the space complexity required in the calculation process increases exponentially as the total number of states increases. When the optimal offline strategy is actually deployed, there is no guarantee that the central controller has enough storage space to store the entire lookup table. It may even be impossible to perform calculations because the state space is too large. Therefore, here we design a new suboptimal online scheduling strategy based on neural network. The idea of this strategy is to replace the lookup table in the previous strategy with a neural network to save storage space. Neural network is a very ideal approximation function of lookup table, theoretically it can be approximated without error. That means in the theory of reinforcement learning, this strategy can achieve the performance of the optimal strategy. We will show that the performance of this suboptimal online strategy is very close to the performance of the optimal offline strategy in the next section.

In order to obtain this neural network, we use a the model-free algorithm called Deep Q Network (DQN). The algorithm continuously learns the hidden laws of the MDP problem by interacting with the environment and continuously trains the neural network to obtain better performance. We show the detailed process of the algorithm in Algorithm 1.

---

#### Algorithm 1: Deep Q Network Algorithm.

---

**Data:** State:  $\mathbb{S} \triangleq \{a', D, \tau, \varphi\}$   
**Result:** Action values:  $A(s, a)$   
 initialization;  
 Initialize data set  $M$ ;  
 Initialize evaluation network  $Q$  with random weights:  $\theta$ ;  
 Initialize target network  $\hat{Q}$  with random weights:  $\theta'$ ;  
**for**  $E = 1 : 2000$  **do**  
   Initialize the environment;  
   Set the origin state  $s_1 \in \mathbb{S}$  randomly;  
   **for**  $t = 1 : 1000$  **do**  
     Choose a random  $a_t$  with probability  $1 - \varepsilon$ ;  
     Otherwise choose  $a_t = \arg \max_a Q(s, a_{t-1}; \theta)$ ;  
     Execute  $a_t$  in the environment;  
     Observe reward  $r_t$  and get new state  $s_{t+1}$ ;  
     Store transition data  $(s_t, a_t, r_t, s_{t+1})$  in  $M$ ;  
     Sample random mini batch of transitions  $(s_j, a_j, r_j, s_{j+1})$  from  $M$ ;  
     Set  $y_j = \begin{cases} r_j & \text{if episode ends at step } j + 1 \\ r_j + \gamma \max_{a'} \hat{Q}(s_{j+1}, a'; \theta') & \text{otherwise} \end{cases}$  ;  
     Perform RMSprop on  $(y_j - Q(s_j, a_j; \theta))^2$  with  $\theta$ ;  
     Every 100 steps, set  $\hat{Q} = Q$   
   **end**  
**end**

---

The structure of the neural network we obtained is shown in Figure 4: Four neurons in the input layer, fifty neurons in the hidden layer, and two neurons in the output layer. This neural network-based scheduling strategy is an online strategy which means that, in order to use this strategy, the current state  $s$  must be input to the neural network first. Then the controller needs to run real-time calculations to obtain the action values  $A(s, a)$  for taking

different actions in the current state. The action value represents how much reward can be obtained by taking the action, so the scheduling strategy is to select the action with the largest  $A(s, a)$  among all actions.

DQN is a relatively mature reinforcement learning algorithm, so we only give the parameter settings of this algorithm and briefly introduce its training process. We run  $E = 2000$  episodes, and each episode contains 1000 steps. In each step, this algorithm executes the greedy strategy with a probability of  $\epsilon = 0.7$ , and the random strategy with a probability of  $1 - \epsilon = 0.3$ . After each step, one state transition datum is stored in the data set. The scale of this data set is  $M = 2048$ , and it is updated in a loop covering manner. A new episode is automatically initialized every 1000 steps. In the meantime, the training process is performed every  $T = 256$  steps, the algorithm selects  $B = 512$  data from the data set for training. The optimizer we used is the Root Mean Square prop optimizer (RMSprop).

With the help of the DQN algorithm, we can obtain the neural network-based suboptimal online strategy. The controller only needs to store the node value of this network, and then calculates the action value in real time according to the current state in each time slot. In other words, this strategy saves a lot of storage space by consuming a small amount of computing ability of the controller. Such an advantage makes this strategy very meaningful in practical applications.

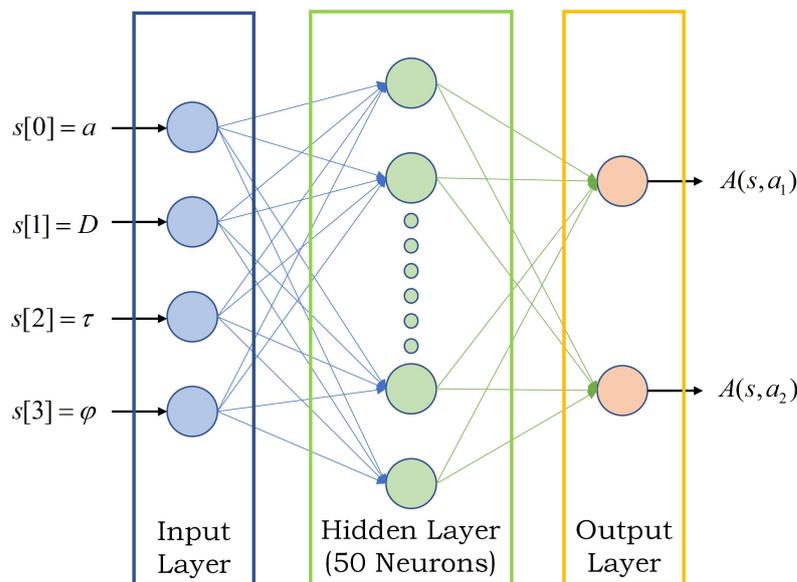


Figure 4. Neural Network Structure.

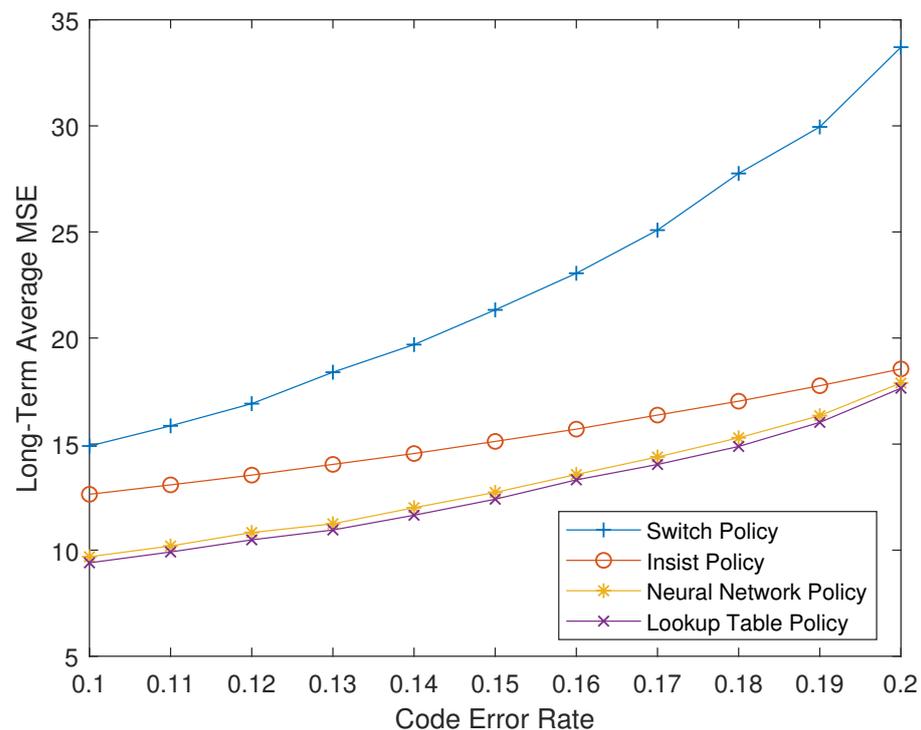
### 5. Numerical Simulation

In this section, we run the numerical simulation on those strategies we proposed and some existing strategies. We illustrate the advantages of the proposed strategies through comparison. First we introduce two benchmark strategies. The first is the switch scheduling strategy, that is, alternate uplink and downlink transmissions between each time slot; the second is the insist scheduling strategy, that is, continuous scheduling of uplink or downlink transmissions until success, then the transmission is exchanged.

The parameter settings in the numerical simulation are as follows: The state transition coefficient is  $A = [1.1, 1.3]$ , the code error rates of the uplink and downlink channels are  $p_s = p_c = [0.1, 0.2]$ , the specific values are marked on the curve obtained from the simulation. The initial state of the plant is  $X_0 = 1$ . The noise distribution is  $\mathcal{N}(\bar{z} = 0, R = 1)$ . The command control coefficient is  $B = -A$ . The initial state control variable is  $s_0 = (a_0, D_0, \tau_0, \varphi_0) = (1, 1, 2, 2)$ . The corresponding initial scheduling action is  $a_0 = 1$ . The initial state of the controller estimation is  $\tilde{X}_0 = 1$ . The range of truncated state space is  $N = \max\{\tau, \varphi\} = 20$ . The plant noise follows normal distribution  $\mathcal{N}(\bar{z} = 0, R = 1)$ . Each

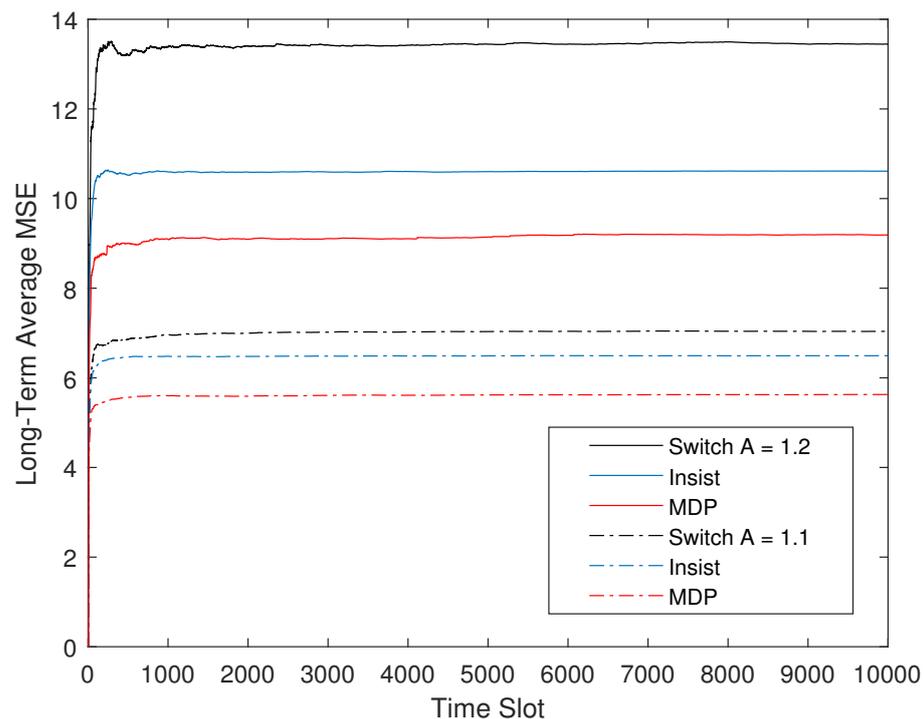
strategy runs 500 episodes with 10,000 time slots each episode. The final long-term average plant state MSE is the average of the results of 500 episodes.

Figure 5 show the long-term average MSE of four strategies with  $A = 1.3$  and  $p_s = p_c = [0.1, 0.2]$ . It can be seen that the MDP strategy, that is, the optimal offline strategy, has the best performance among all strategies, which also is the best performance that all possible scheduling strategies can achieve. While the performance of the neural network-based online strategy has slightly decreased, it is still significantly ahead of the existing strategies, and the performance gap between the optimal offline strategy and the suboptimal online strategy is very small. This gap can be eliminated in theory, but due to the limitations of deep reinforcement learning technology, it is currently difficult to fully achieve the optimal performance. It is relatively simple to obtain a suboptimal strategy with very close performance.



**Figure 5.** Long-term average plant state MSE of four policies with  $A = 1.3$  and  $p_s = p_c = [0.1, 0.2]$ .

Figure 6 show the performance comparison between the optimal offline strategy and the two existing strategies under different state transition coefficient  $A$ . The suboptimal online strategy is not shown because it has been explained that the suboptimal strategy can theoretically approach the optimal. The state transition coefficient and the channel code error rates both reflect the instability of the control system and the reliability of the communication system in Equation (19). Combined with Figure 5, it can be seen that their influence on CPS is the same. A larger state transition coefficient or a higher channel code error rate lead to an increase in the long-term average plant state MSE, and when they exceed a certain limit and no longer satisfy Equation (19), the long-term average MSE of the CPS no longer converges, which means the single-loop CPS is unstable.



**Figure 6.** Long-term average plant state MSE of three policies with  $p_s = p_c = 0.2$  and  $A = \{1.1, 1.2\}$ .

## 6. Conclusions

We proposed the semi-predictive framework to design scheduling strategies for single-loop CPS with uplink and downlink propagation delay. This framework can obtain the optimal offline strategy which is the upper bound on the performance among all strategies and a suboptimal online strategy with more practical application value. By adjusting the parameters, the semi-predictive framework can meet the need of any practical applications. We introduced the complete process of designing scheduling strategies under this framework by taking a specific situation as an example. The numerical simulation proved that the obtained strategies can effectively improve the performance of the existing strategies.

**Author Contributions:** Conceptualization, Z.A. and S.W.; methodology, Z.A. and T.L.; software, Z.A. and J.J.; validation, Z.A., S.W. and Q.Z.; formal analysis, Z.A. and T.L.; investigation, Q.Z.; resources, Z.A.; data curation, Z.A. and S.W.; writing—original draft preparation, Z.A.; writing—review and editing, Z.A.; visualization, Z.A.; supervision, Z.A.; project administration, Z.A.; funding acquisition, S.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded in part by the National Key Research and Development Program of China under Grant no. 2020YFB1806403, and in part by the National Natural Science Foundation of China under Grant nos. 61871147, 61831008, 62071141, 61371102, and in part by the Guangdong Science and Technology Planning Project under Grant no. 2018B030322004.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** Z.A., S.W., T.L., J.J. and Q.Z. would like to thank Zehua Wang, Weihao Guo, Xiao Liang, Jiabao Kang and Dongrui Li for their fruitful insights and discussions.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. Construction Rules of the State Transition Probability Matrix

Here we give the complete construction rules of the state transition matrix. Firstly, we give all the possible new states after a state transition as follows:

$$s'_1 = (1, 2, \tau + 1, \varphi + 1) \tag{A1}$$

$$s'_2 = (2, 2, \tau + 1, \varphi + 1) \tag{A2}$$

$$s'_3 = (2, 0, \tau + 1, \varphi + 1) \tag{A3}$$

$$s'_4 = (1, 2, 1, \varphi + 1) \tag{A4}$$

$$s'_5 = (2, 2, 1, \varphi + 1) \tag{A5}$$

$$s'_6 = (2, 0, 1, \varphi + 1) \tag{A6}$$

$$s'_7 = (1, 1, \tau + 1, \tau + 1) \tag{A7}$$

$$s'_8 = (2, 1, \tau + 1, \tau + 1) \tag{A8}$$

Secondly, we use  $R$  and  $R'$  to mark the transmission results.  $R$  represents the result of the downlink transmission scheduled in the next time slot.  $R'$  represents the result of the uplink transmission arrived in the next time slot. Note that  $R$  is known by prediction while  $R'$  is known by normal communication process. These abbreviations can help to simplify the expression of the rules.

We will give the construction rules in the form of  $P[s'|s, c] = p$  which means that when the condition  $c$  is satisfied, the previous state  $s$  transfers to the new state  $s'$  with a probability of  $p$ .

When  $s = (1, 1, \tau, \varphi)$ :

$$\begin{aligned} P[s'_1|s, a = 1] &= 1 \\ P[s'_2|s, a = 2] &= 1 \end{aligned} \tag{A9}$$

When  $s = (2, 0, \tau, \varphi)$ :

$$\begin{aligned} P[s'_7|s, a = 1] &= 1 \\ P[s'_8|s, a = 2] &= 1 \end{aligned} \tag{A10}$$

When  $s = (2, 1, \tau, \varphi)$ :

$$\begin{aligned} P[s'_1|s, a = 1] &= 1 \\ P[s'_2|s, a = 2] &= 1 \end{aligned} \tag{A11}$$

When  $s = (2, 2, \tau, \varphi)$ :

$$\begin{aligned} P[s'_1|s, a = 1] &= 1 \\ P[s'_2|s, a = 2, R = 0] &= p_c \\ P[s'_2|s, a = 2, R = 1, \tau = \varphi] &= p_s \cdot (1 - p_c) \\ P[s'_3|s, a = 2, R = 1, \tau \neq \varphi] &= p_s \cdot (1 - p_c) \end{aligned} \tag{A12}$$

When  $s = (1, 2, \tau, \varphi)$ :

$$\begin{aligned} P[s'_1|s, a = 1] &= p_s \\ P[s'_2|s, a = 2, R' = 0, R = 0] &= p_s \cdot p_c \\ P[s'_2|s, a = 2, R' = 0, R = 1, \tau = \varphi] &= p_s \cdot (1 - p_c) \\ P[s'_3|s, a = 2, R' = 0, R = 1, \tau \neq \varphi] &= p_s \cdot (1 - p_c) \\ P[s'_4|s, a = 1, R' = 1] &= 1 - p_s \\ P[s'_5|s, a = 2, R' = 1, R = 0] &= (1 - p_s) \cdot p_c \\ P[s'_6|s, a = 2, R' = 1, R = 1] &= (1 - p_s) \cdot (1 - p_c) \end{aligned} \tag{A13}$$

### Appendix B. Proof of Theorem 1

#### Appendix B.1. Scheduling 1 Subsystem per Time Slot without Delay

To prove sufficient conditions, we only need to prove that there exists a stationary deterministic strategy that can make multi-loop CPS stable. Here we prove that the round-robin insist scheduling strategy can keep the system stable. We first prove the case of  $L = 1$ . Round-robin means that in every  $K$  time slots, the controller schedules each subsystem

once in turn, and the scheduling sequence is fixed from  $i = 1$  to  $i = K$ . Insist refers to when scheduling each subsystem, continuously scheduling uplink or downlink transmission until it succeeds, then switch to another transmission. Therefore, the actions of a single subsystem under the round-robin insist scheduling strategy can be given in the form of the following time axis:

The time axis between two consecutive successful downlink scheduling is recorded as a control loop. It can be seen that the AoI evolution process of each control loop of one subsystem is:

- (1) The initial estimation age is equal to the control age:  $n'K$
- (2) The current subsystem waits for the completion of the scheduling of other subsystems, that is, silence  $(k - 1)$  time slots, and then schedules the uplink transmission when it is scheduled again. If the uplink transmission fails, the subsystem waits another  $(k-1)$  time slots and tries again until the uplink transmission is successful. This step takes  $mK$  time slots. At the end of this step, the estimated age is 0, and the control age is  $(n' + m)K$ ;
- (3) After the current subsystem silences for  $(K - 1)$  time slots, it switches to schedule downlink transmission continuously until it succeeds. This step takes  $nK$  time slots. At the end of this step, the estimated age is equal to the control age:  $nK$ . Then it finishes a close control loop.

Note that the time slots included in a complete control loop are the time slots marked in red on the coordinate axis in Figure A1, that is, the control age ranges from  $n'K$  to  $(n' + m + n)K$ . Each control loop has repeatability, so we only need to prove that the long-term average cost within the range of one control loop converges.

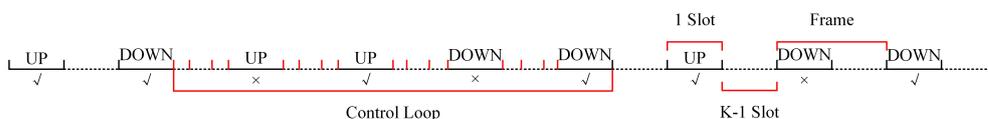


Figure A1. The round-robin insist scheduling strategy.

According to the channel error probability, the  $M$  uplink transmissions and  $N$  downlink transmissions in each control loop can be modeled as a geometric distribution with the probability of success being  $(1 - p_s)$  and  $(1 - p_c)$  respectively.  $M$  and  $N$  are different in each control loop,  $N'$  Represents the number of downlink transmissions in the previous loop of the current control loop.  $(n', m, n)$  are their specific observations.  $C_i$  and  $T_i$  represent the total cost and total time of the  $i$ -th control loop of the current subsystem respectively:

$$C_i = \sum_{q=0}^{(m+n)K-1} f(n'K + q) = \sum_{q=1}^{(m+n)K} f(n'K + q - 1) \tag{A14}$$

$$T_i = (m + n)K \tag{A15}$$

where  $f(\varphi) = \sum_{i=1}^{\varphi} (A^2)^{i-1}$ . Next, we can express the long-term average cost as:

$$J = \lim_{t \rightarrow \infty} \frac{C_1 + C_2 + \dots + C_t}{T_1 + T_2 + \dots + T_t} = \frac{\mathbb{E}[C]}{\mathbb{E}[T]} \tag{A16}$$

$$\mathbb{E}[C] = \sum_{n'} \sum_m \sum_n \left( \mathbb{E}[C|N' = n', M = m, N = n] \cdot \mathbb{P}[N' = n', M = m, N = n] \right) \tag{A17}$$

$$E[L] = \sum_{n'} \sum_m \sum_n ((m + n) \cdot K \cdot \mathbb{P}[N' = n', M = m, N = n]) \tag{A18}$$

It can be seen that if  $\mathbb{E}[C]$  is bounded, then  $J$  is bounded. According to the definition of  $C_i$  and the three geometrical distributions  $(N', M, N)$ , which are independent of each other, we have:

$$\mathbb{E}[C|N' = n', M = m, N = n] = \sum_{q=0}^{(m+n)K-1} f(n'K + q) = \sum_{q=1}^{(m+n)K} f(n'K + q - 1) \tag{A19}$$

$$\begin{aligned} & \mathbb{P}[N' = n', M = m, N = n] \\ &= \mathbb{P}[N' = n'] \cdot \mathbb{P}[M = m] \cdot \mathbb{P}[N = n] \\ &= (1 - p_c)p_c^{n'-1}(1 - p_s)p_s^{m-1}(1 - p_c)p_c^{n-1} \end{aligned} \tag{A20}$$

Choose  $p_{\max} = \max\{p_s, p_c\}$ , we can derive that:

$$\mathbb{E}[C] \leq \alpha_1 \cdot \sum_{n'} \sum_m \sum_n \left( \sum_{q=1}^{(m+n)K} f(n'K + q - 1) \cdot p_{\max}^{n'+m+n} \right) \tag{A21}$$

where  $\alpha_1 = (1 - p_c)p_c^{-1}(1 - p_s)p_s^{-1}(1 - p_c)p_c^{-1}$ . Since  $f(\cdot)$  is a strictly increasing function and  $(n', m, n)$  are all greater than 0, we can derive that:

$$\mathbb{E}[C] < \alpha_2 \cdot \sum_{n'} \sum_m \sum_n \left( (n' + m + n) \cdot f(n'K + mK + nK) \cdot p_{\max}^{n'+m+n} \right) \tag{A22}$$

where  $\alpha_2 = K(1 - p_c)p_c^{-1}(1 - p_s)p_s^{-1}(1 - p_c)p_c^{-1}$ . We abbreviate  $n' + m + n$  as  $i$ , that is,  $i = n' + m + n$ . Considering  $i \geq 3$ , and when  $i = n' + m + n$  is a fixed value, the possible combinations of  $(n', m, n) \geq 1$  satisfy the mathematical relationship of  $\sum_{n'} \sum_m \sum_n (1) < (n' + m + n)^3$ , namely:

$$\sum_{n'} \sum_m \sum_n (n' + m + n) < (n' + m + n)^3 \cdot (n' + m + n) \tag{A23}$$

$$\sum_{n'} \sum_m \sum_n (i) < (i)^4 \tag{A24}$$

We can derive that:

$$\mathbb{E}[C] < \alpha_2 \cdot \sum_i \left( i^4 \cdot f(iK) \cdot p_{\max}^i \right) \tag{A25}$$

Since there are always exist  $p > p_{\max}$  and  $n < \infty$ , satisfying  $i^4 p_{\max}^i < p^i, \forall i > n$ . So we have:

$$\sum_i \left( i^4 \cdot f(iK) \cdot p_{\max}^i \right) < \sum_i \left( f(iK) \cdot p^i \right) \tag{A26}$$

So if  $\sum_i (f(iK) \cdot p^i) < \infty$ , then  $\sum_i (i^4 \cdot f(iK) \cdot p_{\max}^i) < \infty$ . Now seeking the conditions for the stability of the multi-loop CPS subsystem is transformed into seeking the conditions for the establishment of  $\sum_i (f(iK) \cdot p^i) < \infty$ . For  $f(iK)$ , we have:

$$f(iK) = \sum_{q=1}^{iK} (A^2)^{q-1} = 1 + A^2 + A^4 + \dots + A^{2(iK-1)} = \frac{1 - (A^2)^{iK}}{1 - A^2} \tag{A27}$$

For  $\sum_i (f(iK) \cdot p^i)$ , we have:

$$\begin{aligned}\sum_i^{\infty} (f(iK) \cdot p^i) &= \sum_i^{\infty} \left( \frac{1 - (A^2)^{iK}}{1 - A^2} \cdot p^i \right) = \frac{1}{1 - A^2} \sum_i^{\infty} \left( (1 - (A^2)^{iK}) \cdot p^i \right) \\ &= \frac{1}{1 - A^2} \left( \sum_i^{\infty} (p^i) - \sum_i^{\infty} ((A^2)^{iK} p^i) \right)\end{aligned}\quad (\text{A28})$$

So in order to ensure that  $\frac{1}{1-A^2} \left( \sum_i^{\infty} (p^i) - \sum_i^{\infty} ((A^2)^{iK} p^i) \right) < \infty$  is satisfied, it is obvious that  $p < 1$  and  $A^{2K}p < 1$  must stand, that is,  $p < \left(\frac{1}{A^2}\right)^K$ . This completes the proof.

### Appendix B.2. Scheduling $L$ Subsystems per Time Slot without Delay

When each time slot can schedule  $L$  subsystems, the corresponding strategy can be set to multiple independent round-robin insist scheduling strategies. It can be ensured that the round-robin cycle of each subsystem does not exceed  $\lceil K/L \rceil$ , and the follow-up proof is consistent with Appendix B.1.

### Appendix B.3. Scheduling $L$ Subsystems per Time Slot with Delay

For a specific subsystem, we assume that the fixed delay for each transmission is  $D_i$  frames, which is equivalent to the uplink and downlink scheduling in the control lcoopf must be delayed by  $D_iK$  time slots for AA reception, so the formula (A19) is modified as follows :

$$\begin{aligned}\mathbb{E}[C|N' = n', M = m, N = n] &= \sum_{q=0}^{(m+n+2D)K-1} f(n'K + DK + q) \\ &= \sum_{q=1}^{(m+n+2D)K} f(n'K + DK + q - 1)\end{aligned}\quad (\text{A29})$$

Since  $\mathbb{E}[D] = \mathbb{E}[D_i] = D$  is a constant which has no effect on the subsequent proof, the proof process is consistent with Appendix B.1.

## References

- Zhang, X.M.; Han, Q.L.; Ge, X.; Ding, D.; Ding, L.; Yue, D.; Peng, C. Networked control systems: A survey of trends and techniques. *IEEE/CAA J. Autom. Sin.* **2020**, *7*, 1–17. [\[CrossRef\]](#)
- Lin, J.; Yu, W.; Zhang, N.; Yang, X.; Zhang, H.; Zhao, W. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet Things J.* **2017**, *4*, 1125–1142. [\[CrossRef\]](#)
- Xu, H.; Yu, W.; Griffith, D.; Golmie, N. A Survey on Industrial Internet of Things: A Cyber-Physical Systems Perspective. *IEEE Access* **2018**, *6*, 78238–78259. [\[CrossRef\]](#)
- Lu, C.; Saifullah, A.; Li, B.; Sha, M.; Gonzalez, H.; Gunatilaka, D.; Wu, C.; Nie, L.; Chen, Y. Real-Time Wireless Sensor-Actuator Networks for Industrial Cyber-Physical Systems. *Proc. IEEE* **2016**, *104*, 1013–1024. [\[CrossRef\]](#)
- Liu, W.; Nair, G.; Li, Y.; Nesic, D.; Vucetic, B.; Poor, H.V. On the Latency, Rate, and Reliability Tradeoff in Wireless Networked Control Systems for IIoT. *IEEE Internet Things J.* **2021**, *8*, 723–733. [\[CrossRef\]](#)
- Han, B.; Zhu, Y.; Jiang, Z.; Hu, Y.; Schotten, H.D. Optimal Blocklength Allocation Towards Reduced Age of Information in Wireless Sensor Networks. In Proceedings of the 2019 IEEE Globecom Workshops (GC Wkshps), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.
- Han, B.; Jiang, Z.; Zhu, Y.; Schotten, H.D. Recursive Optimization of Finite Blocklength Allocation to Mitigate Age-of-Information Outage. In Proceedings of the 2020 IEEE International Conference on Communications Workshops (ICC Workshops), Dublin, Ireland, 7–11 June 2020; pp. 1–6.
- Li, D.; Wu, S.; Wang, Y.; Jiao, J.; Zhang, Q. Age-Optimal HARQ Design for Freshness-Critical Satellite-IoT Systems. *IEEE Internet Things J.* **2020**, *7*, 2066–2076. [\[CrossRef\]](#)
- Parag, P.; Taghavi, A.; Chamberland, J. On Real-Time Status Updates over Symbol Erasure Channels. In Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA, 19–22 March 2017; pp. 1–6.

10. Huang, K.; Liu, W.; Li, Y.; Savkin, A.; Vucetic, B. Wireless Feedback Control with Variable Packet Length for Industrial IoT. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 1586–1590. [[CrossRef](#)]
11. Liu, C.-F.; Bennis, M. Data-Driven Predictive Scheduling in Ultra-Reliable Low-Latency Industrial IoT: A Generative Adversarial Network Approach. In Proceedings of the 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Atlanta, GA, USA, 26–29 May 2020; pp. 1–5.
12. Eisen, M.; Gatsis, K.; Pappas, G.J.; Ribeiro, A. Learning in Wireless Control Systems Over Nonstationary Channels. *IEEE Trans. Signal Process.* **2019**, *67*, 1123–1137. [[CrossRef](#)]
13. Sun, Y.; Kadota, I.; Talak, R.; Modiano, E. *Age of Information: A New Metric for Information Freshness*; Morgan & Claypool: San Rafael, CA, USA, 2019.
14. Sinopoli, B.; Schenato, L.; Franceschetti, M.; Poolla, K.; Jordan, M.I.; Sastry, S.S. Kalman filtering with intermittent observations. *IEEE Trans. Autom. Control* **2004**, *49*, 1453–1464. [[CrossRef](#)]
15. Champati, J.P.; Mamduhi, M.H.; Johansson, K.H.; Gross, J. Performance Characterization Using AoI in a Single-loop Networked Control System. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, 29 April–2 May 2019; pp. 197–203.
16. Park, P.; Araújo, J.; Johansson, K.H. Wireless networked control system co-design. In Proceedings of the 2011 International Conference on Networking, Sensing and Control, Delft, The Netherlands, 11–13 April 2011; pp. 486–491.
17. Huang, K.; Liu, W.; Li, Y.; Vucetic, B. To Retransmit or Not: Real-Time Remote Estimation in Wireless Networked Control. In Proceedings of the ICC 2019—2019 IEEE International Conference on Communications (ICC), Shanghai, China, 21–23 May 2019; pp. 1–7.
18. An, Z.; Wu, S.; Wang, Y.; Jiao, J.; Zhang, Q. HARQ Based Joint Uplink-Downlink Optimal Scheduling Strategy for Single-Loop WNCS. In Proceedings of the 2020 International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 21–23 October 2020; pp. 7–12.
19. Huang, K.; Liu, W.; Li, Y.; Vucetic, B.; Savkin, A. Optimal Downlink—Uplink Scheduling of Wireless Networked Control for Industrial IoT. *IEEE Internet Things J.* **2020**, *7*, 1756–1772. [[CrossRef](#)]
20. Wang, X.; Chen, C.; He, J.; Zhu, S.; Guan, X. AoI-Aware Control and Communication Co-design for Industrial IoT Systems. *IEEE Internet Things J.* **2020**, *8*, 8464–8473. [[CrossRef](#)]
21. Jiang, Z.; Krishnamachari, B.; Zhou, S.; Niu, Z. Can Decentralized Status Update Achieve Universally Near-Optimal Age-of-Information in Wireless Multiaccess Channels? In Proceedings of the 2018 30th International Teletraffic Congress (ITC 30), Vienna, Austria, 3–7 September 2018; pp. 144–152.
22. Ayan, O.; Vilgelm, M.; Kellerer, W. Optimal Scheduling for Discounted Age Penalty Minimization in Multi-Loop Networked Control. In Proceedings of the 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 10–13 January 2020; pp. 1–7.
23. Girgis, A.M.; Park, J.; Liu, C.-F.; Bennis, M. Predictive Control and Communication Co-Design: A Gaussian Process Regression Approach. In Proceedings of the 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Atlanta, GA, USA, 26–29 May 2020; pp. 1–5.
24. Chang, B.; Zhang, L.; Li, L.; Zhao, G.; Chen, Z. Optimizing Resource Allocation in URLLC for Real-Time Wireless Control Systems. *IEEE Trans. Veh. Technol.* **2019**, *68*, 8916–8927. [[CrossRef](#)]
25. Maity, D.; Mamduhi, M.H.; Hirche, S.; Johansson, K.H.; Baras, J.S. Optimal LQG Control Under Delay-Dependent Costly Information. *IEEE Control Syst. Lett.* **2019**, *3*, 102–107. [[CrossRef](#)]
26. Mamduhi, M.H.; Maity, D.; Baras, J.S.; Johansson, K.H. A Cross-Layer Optimal Co-Design of Control and Networking in Time-Sensitive Cyber-Physical Systems. *IEEE Control Syst. Lett.* **2021**, *5*, 917–922. [[CrossRef](#)]
27. Maity, D.; Baras, J.S. Minimal Feedback Optimal Control of Linear-Quadratic-Gaussian Systems: No Communication is also a Communication. *IFAC-PapersOnLine* **2020**, *53*, 2201–2207. [[CrossRef](#)]
28. Schenato, L.; Sinopoli, B.; Franceschetti, M.; Poolla, K.; Sastry, S.S. Foundations of Control and Estimation Over Lossy Networks. *Proc. IEEE* **2007**, *95*, 163–187. [[CrossRef](#)]