Research Article

A Time-Delay-Bounded Data Scheduling Algorithm for Delay Reduction in Distributed Networked Control Systems

Zeming Fan,1 Xiaojun Yu,1 Hao Wan,1 Meilin Kang,1 Yuan Liu,1 Yuye He,1 and Gaoxi Xiao2

1School of Automation, Northwestern Polytechnical University, Xi’an, Shaanxi 710129, China
2School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798

Correspondence should be addressed to Xiaojun Yu; e070035@e.ntu.edu.sg

Received 5 February 2020; Revised 19 May 2020; Accepted 9 June 2020; Published 21 July 2020

Guest Editor: Sahar Yazdani

Copyright © 2020 Zeming Fan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As a key feature of networked control systems (NCSs), the time delays induced by communication medium sharing and data exchange over the system components could largely degrade the NCS performances or may even cause system instability, and thus, it is of critical importance to reduce time delays within NCSs. This paper studies the time-delay reduction problem in distributed NCSs and presents a dual-way data scheduling mechanism for time-delay reductions in delay-bounded NCSs with time-varying delays. We assess the time delays and their influences on the NCSs first with various delay factors being considered and then describe a one-way scheduling mechanism for network-delay reductions in NCSs. Based upon such a method, a dual-way scheduling algorithm is finally proposed for distributed NCSs with different types of transmitted data packets. Experiments are conducted on a remote teaching platform to verify the effectiveness of the proposed dual-way scheduling mechanism. Results demonstrate that, with the stability time-delay bound considered within the scheduling process, the proposed mechanism is effective for NCS time-delay reductions while addressing the stability, control accuracy, and settling time issues efficiently. Such a proposed mechanism could also be implemented together with some other existing control algorithms for time-delay reductions in NCSs. Our work could provide both useful theoretical guidance and application references for stable tracking control of delay-bounded NCSs.

1. Introduction

With the rapid advancements of networking technologies over the past decades, there is a growing trend in both industrial and commercial communities to integrate computing, communication, and control systems together to formulate a unified platform via network remote control. In such integrated platforms, the control commands from different information sources are transmitted and exchanged over networks, while their feedback control systems, whose control loops are formed via real-time communication channels, are called networked control systems (NCSs) [1]. Owing to its advantageous properties, e.g., low cost, high reliability, easy reconfiguration, satisfactory flexibility, robustness, and adaptation capabilities, NCS has attracted extensive research interest in recent years and been utilized in various fields, such as the power grids, transportation networks, water distribution networks, telephone networks, global financial networks, and genetic expression networks [2].

In practical NCSs, there exist numerous information sources and nodes exchanging data over the networks simultaneously, and thus, the time delays induced by network resource sharing would generate inevitably [3–5]. Such time delays could largely degrade the system performances or sometimes may even cause the system to be instable. Furthermore, as the time delays within NCSs could either be constant, or bounded, or random, they also make the NCS system design and analysis complicated. Therefore, it is of critical importance to reduce the NCS time delays in engineering practice [5, 6].

Various mechanisms have been proposed for NCS time-delay reductions in the literature [7, 8]. Based on the
different time-delay models being adopted, such mechanisms could be divided into two main categories, namely, the NCS network control-based ones and the NCS system data scheduling-based ones [7]. Specifically, the NCS network control-based mechanisms treat the whole NCS system as a controlled network, and the main focus is on control of the networks, i.e., the main object is to propose efficient methods to address the raised networking issues, such as routing, congestion control, networking protocol, and data communications [6–8]. The NCS network control-based mechanisms have attracted extensive research interest over the past years, and readers could refer to [6–8] for comprehensive details.

The data scheduling-based mechanisms consider NCSs to be control systems and try to optimize performances of the overall NCSs [9, 10]. Initially, the data scheduling-based mechanisms assume that the NCSs have fixed data sampling period and propose methods to optimize resource allocations for competing demands [10]. Scheduling algorithms such as rate monotony (RM) scheduling, earliest deadline first (EDF) dynamic scheduling, and dead-bands scheduling, are of this category. Although such methods help reduce time delays, the practical system variables, such as varying transmission delays, transmission variables, packet loss, and communication constraints, have been ignored, and thus, these methods may not be adaptive to real-time network changes during the system operation process [11]. To further overcome these problems, scheduling mechanisms with variable sampling periods, such as dynamic feedback scheduling, fuzzy logic control or neural network-based scheduling, active sampling period scheduling, and delay compensation-based scheduling, have also been proposed [11–14]. Those methods are flexible in handling the time-varying traffic, yet some other constraints, such as the unknown network loads prior to scheduling, the lacks of upper and lower bounds for sampling period, and the difficulties in determining the key control parameters, have been largely ignored [14, 15].

Event-triggering feedback control mechanism is another kind of scheduling scheme being proposed to address the above network constraints and has attracted extensive research interest in recent years [16–18]. Specifically, by adopting a subsystem to broadcast the local state information to its neighbors, the distributed event-triggering feedback schemes were proposed for linear and nonlinear systems first [16, 17], and later, such algorithms were extended with various practical constraints, e.g., the data drops, transmission delays, probabilistic nonlinearities, sensor/actuator faults, and external attacks/disturbances, being taken into account, and have also achieved satisfactory results [18–21]. Currently, some other algorithms with more practical constraints taken into account are attracting increasing research interest. For more details of recent advances on event-triggering NCS control and the trends and techniques for NCS delay reductions, readers could refer to [6, 22]. It is worth mentioning that most of those existing mechanisms have ignored the influences of the upper and lower time-delay bounds on the performances of the system stability and the data scheduling schemes.

In practice, however, the permissible system time-delay bounds play a critical role in determining the system stability performances for NCSs. This is because, on the one hand, there usually exists a huge amount of information being exchanged over the networks, and such information would inevitably induce time delays, while on the other hand, such information is typically of different types, and each type of information may have its own transmitting priorities within the NCS. For such information being transmitted, once the transmission time exceeds the system permissible time-delay bounds, the system may become unstable or even divergent [23]. The case is especially true for distributed NCSs, since in decentralized systems, some of the plants are first-ordered, while the others may be higher-ordered with much stricter time-delay bound limitations. Therefore, to avoid causing system instability, scheduling in NCSs should be performed within the system time-delay bounds.

It is also worth noting that most of the existing scheduling mechanisms are one-way designed, making the scheduling operations inefficient for NCS delay reductions. For instance, for distributed NCSs as shown in Figure 1, those existing scheduling mechanisms typically operate either on the field sensor scheduler or on the network remote controller scheduler only. In practice, however, if the scheduling operations are conducted on the sensor scheduler and controller schedulers simultaneously, i.e., dual-way scheduling schemes are devised and adopted, the NCS time-delay reductions could be much more efficient. This is because, in such a case, both sensor terminals and remote controller could transmit their control demands simultaneously within the NCS for processing, and meanwhile, the sensor terminals and remote controllers could work cooperatively and more efficiently with the control variables being transmitted within the NCS control loops.

This paper studies the time-delay reduction problem within distributed NCSs and proposes a delay-bounded data scheduling mechanism for time-delay reductions in NCSs. Specifically, with the various delay factors being considered, the time delays and their influences on the NCSs are evaluated first, and then, a one-way scheduling algorithm is presented for data scheduling within NCSs. Based on such a one-way scheduling scheme, a novel dual-way dynamic scheduling algorithm, which is performed on controller and sensor schedulers simultaneously, is finally described. Both simulations and experiments are conducted to verify the effectiveness of the proposed data scheduling mechanism. Results obtained from experiments carried out on a practical remote teaching platform show that, with the scheduling operations conducted within the lower and upper NCS delay bounds, the proposed algorithm could help significantly improve the performances of the distributed NCSs while the system stability could also be guaranteed in different cases.

2. Time Delays and Their Impacts on Distributed NCSs

2.1. Time Delays within NCSs. A typical NCS comprised several control plants, sensors, controllers, and actuators shown in Figure 2. In such an NCS, there are numerous
information sources sharing the network resources and transmitting data onto the network via communication channels. Once too much information are transmitted simultaneously via the same channel, the common network transmission issues, e.g., congestions, packet collisions, multipath transmissions, and link interruptions, may arise and thus would cause time delays inevitably [3, 4].

The time delays within NCSs could be categorized into control-induced delays and network-induced delays. While the former is the time consumed by the sensors, plants, and actuators to complete their respective noncommunication functions, the latter is induced by data transmissions within the NCS. For NCSs with feedback and forward channels as shown in Figure 1, we further divide the time delays in the system $k$th control cycle into five parts as follows:

1. Data preprocessing delay, i.e., the time required by sensors to pack data for transmission, and it is denoted by $T_{prc}^{\text{src}}(k)$ and $T_{prc}^{\text{src}}(k)$ for the feedback and forward channels, respectively.
2. Data packet queuing delay, i.e., the time taken by a data packet to wait for its transmission, and it is denoted by $T_{wait}^{\text{src}}(k)$ and $T_{wait}^{\text{src}}(k)$ for the feedback and forward channels, respectively.
3. Data transmission delay, i.e., the time taken by a data packet to be transmitted within the system, which is determined by the packet length, network bandwidth, and transmission distance. Such a delay is denoted by $T_{tra}^{\text{src}}(k)$ and $T_{tra}^{\text{src}}(k)$ for the feedback and forward channels, respectively.
4. Data postprocessing delay, i.e., the time required for a controller to receive and store a data packet, and it is denoted by $T_{pos}^{\text{src}}(k)$ and $T_{pos}^{\text{src}}(k)$ for the feedback and forward channels, respectively.
5. Controller calculation delay is denoted by $T_{c}^{\text{src}}(k)$.

![Figure 1: Structure of a typical NCS with time delays.](image1)

![Figure 2: Topological structure of network with $n$ controllers.](image2)
Denote the overall time delay for the feedback and forward channels to be $T_{ii}(k)$ and $T_{i2}(k)$; then, we can have

$$T_{ii}(k) = T_{ii}^{prc}(k) + T_{i2}^{wait}(k) + T_{i2}^{tra}(k) + T_{ii}^{pos}(k),$$

$$T_{i2}(k) = T_{i2}^{prc}(k) + T_{i2}^{wait}(k) + T_{i2}^{tra}(k) + T_{i2}^{pos}(k),$$

while the total time delay within the NCS is

$$T_i(k) = T_{ii}(k) + T_{i2}(k) + T_i^p(k),$$

which is also the control system round trip time (RTT), i.e., the interval between the time a plant receives its $k$th control command to the time when it receives its $(k + 1)$th command.

In practice, $T_i^p(k)$ is determined by computer performances. Specifically, since such a delay is much smaller as compared with the others and could be compensated by algorithms [24], the influences of $T_i^p(k)$ are neglected and only those of $T_{ii}(k)$ and $T_{i2}(k)$ are evaluated in this paper.

2.2. Influences of Time Delays on NCS Stability. Assume that controllers in the NCS as shown in Figure 2 are event-driven, while those sensors and actuators are time-driven, then the theoretical discrete state equations for the $i$th plant are

$$x_i(k + 1) = A_i x_i(k) + B_i u_i(k),$$

$$u_i(k) = K_i x_i(k),$$

where $x_i(k) \in \mathbb{R}^n$ is the state variable of the $i$th plant in its $k$th control cycle; $A_i$ and $B_i$ are constant matrices, while $u_i(k)$ and $K_i$ are control input and feedback gain of the $i$th control plant, respectively.

Further assume that the network-induced delays are time-varying and bounded, satisfying [25, 26]

$$0 \leq T_i(k) \leq T_{iM},$$

where $T_{iM}$ is a constant, denoting the upper bound of the time-varying delay $T_i(k)$, and it could be characterized by the summation inequality presented in Theorem 5 in [27].

With the network-induced delays being taken into account, we have $u_i(k) = K_i x_i(k - T_i(k))$ from (5), while the closed-loop NCS by (4) and (5) could be described as follows:

$$x_i(k + 1) = A_i x_i(k) + B_i K_i x_i(k - T_i(k)) = A_i x_i(k) + A_{id} x_i(k - T_i(k)),$$

$$y_i(k) = c_i x_i(k),$$

where $A_{id} = B_i K_i$ is a loose variable introduced by Lyapunov’s function to characterize the NCS. The schematic diagram of such a control system could be illustrated by Figure 3, wherein the two networks could either be the same or different depending on the practical implementation of the NCSs.

3. Strategies for Network-Delay Reductions

As shown in (1), the time delay $T_{ii}(k)$ in NCS is caused mainly by the data processing delay $T_{ii}^{prc}(k)$, queuing delay $T_{i2}^{wait}(k)$, transmission delay $T_{i2}^{tra}(k)$, and the postprocessing delay $T_{ii}^{pos}(k)$. Among all those factors, since $T_{ii}^{prc}(k)$ is determined by the network conditions, while $T_{i2}^{wait}(k)$ and $T_{ii}^{pos}(k)$ are so short that could be negligible [28], $T_{i2}^{tra}(k)$ is regarded to be the main delay factor. This is similar for $T_{i2}(k)$ in the forward channels. However, since the data transmitted in NCS system include video/audio information, control commands, sensing data, and control data of all NCS components, which are either periodic or nonperiodic and to be transmitted either in real time or non-real time over the channels, the NCS time delays could be very large. In such a case, the transmission of all control data or system outputs within a single data packet is impractical. Moreover, since the queuing delays of both forward and backward channels could be manipulated by scheduling of the sensors and controllers, it is expected that appropriate scheduling algorithms could be devised to minimize the system time delays.

In this paper, we propose a two-way scheduling mechanism for an NCS, wherein scheduling operations are performed on both sensor and controller schedulers. Specifically, once there exist any data collisions, the scheduling mechanism is adopted to assign the data packets to each node different priorities, such that they could be transmitted with the shortest average delay.

3.1. Scheduling Preprocessing. To describe the scheduling operations for a plant within a control loop more clearly, we categorize the scheduling operations into data preprocessing on sensor schedulers, data postprocessing on controller schedulers, data queuing on controller schedulers, and data queuing on sensor schedulers. Hence, the time delays for each plant are mainly introduced by these four tasks and could be managed by a scheduling operation. Specifically, for any plant $i$, a transmission description function $t_{in}$ could be established as shown in (8), which could be utilized to calculate the desired scheduling time slot for each data packet.

$$t_{in} = f(t_{in}, t_{in}, t_{in}, t_{in}, t_{in}, t_{in}),$$

where $n = 1, 2, 3, 4$ denoting the four data packet scheduling operations. Hence, the time delays as shown in Figure 1 could also be denoted as $T_{i2}^{wait} = t_{i1}$, $T_{i2}^{pos} = t_{i2}$, $T_{i2}^{pos} = t_{i3}$, and $T_{i2}^{pos} = t_{i4}$. $t_{in}$ describes the significance level of each data packet and is defined to be the reciprocal of the packet
time-delay upper bound. \( \tau_{\text{ina}} \) is the actual packet transmission starting time, which is also utilized to order the scheduling operations for packets with the same significance level. \( \tau_{\text{inr}} \) is the longest execution time among all data packets, which is determined by the data packet length, network bandwidth, and transmission distances. \( \tau_{\text{inf}} \) and \( \tau_{\text{ine}} \) are the earliest and latest transmission starting time of a data packet, respectively. \( \tau_{\text{ind}} \) is the control cycle of the NCS.

Once the system delay bounds \( T_{i1M}, T_{i2M} \) and the packet transmission description function \( \tau_{\text{in}} \) are determined, the scheduling operations then could be decided for each packet in the NCS. Specifically, for a packet to be transmitted, its scheduling operation includes two main steps, i.e., packet significance level determination and scheduling operation determination for all packets with the same significance level. In this study, we call the former as scheduling preprocessing and the latter as packet scheduling operation. The determinations of packet transmission property variables are discussed in the following sections.

3.2. Scheduling Mechanism for Sensor Scheduler

3.2.1. Scheduling of Coupled Information. In NCSs with coupled information, data packets are transmitted in predefined order, i.e., for a plant within a control loop, its control information should be sent to the actuator before it collects data from a sensor, while for a controller, it has to receive sensor data first, and then sends the control information to the local controller via network after executing the control mechanism. In this way, the real-time data transmission within NCSs could be guaranteed. To perform scheduling for packets with the same significance level, a data transmission set \( \tau \) has to be established as below according to the packet transmission time, while each packet should be assigned a time slot, within which the sensor scheduler or controller scheduler is fully occupied by the scheduling task specified:

\[
\tau_{\text{s}} = \left\{ \tau_{\text{in}} \to \tau_{\text{jn}} \middle| \tau_{\text{in}}, \tau_{\text{jn}} \in \tau \right\},
\]

where \( \tau \) represents all data packets to be transmitted, and \( [\tau_{\text{in}} \to \tau_{\text{jn}}] \) defines the relative priority of two data packets within \( \tau \), indicating that \( \tau_{\text{jn}} \) can be transmitted only when the transmission of \( \tau_{\text{in}} \) is completed.

However, since the processing of a packet may last for a period of time, the variables \( \tau_{\text{in}}, \tau_{\text{inf}}, \tau_{\text{ine}} \) have to be calculated. For an existing data transmission set with \( u \) packets \( [\tau_{\text{in}} \to \tau_{\text{jn}}] (k = 1, \ldots, u) \) to be transmitted, the earliest transmission starting time \( \tau_{\text{inf}}' \) among those packets can be calculated as follows:

\[
\tau_{\text{inf}}' = \tau_{\text{ina}} + \tau_{\text{inr}},
\]

where \( \tau_{\text{ina}} = T_{i1M}^{\text{tra}} \). The earliest transmission starting time \( \tau_{\text{inf}} \) among all packets within the set could be calculated:

\[
\tau_{\text{inf}} = \max \left( \tau_{\text{inf}}', \ldots, \tau_{\text{inf}}'^{u} \right),
\]

Similarly, if there exists a transmission set with \( v \) packets following \( [\tau_{\text{in}} \to \tau_{\text{in}}'] (k = 1, \ldots, v) \), then the latest transmission starting time \( \tau_{\text{ine}}' \) for those data packets can be calculated as follows:

\[
\tau_{\text{ine}}' = \min \left( \tau_{\text{ine}} - \tau_{\text{inr}}, T_{i1M} - \tau_{\text{inr}} \right),
\]

and the latest transmission starting time \( \tau_{\text{ine}} \) for packets within the transmission set could be determined by

\[
\tau_{\text{ine}} = \min \left( \tau_{\text{ine}}', \tau_{\text{ine}}'^{v} \right).
\]

3.2.2. Scheduling of Noncoupled Information. In NCSs, there also exists noncoupled information with the packets being transmitted in an arbitrary order. To determine the scheduling operations for such packets, the packet transmission starting time \( \tau_{\text{in}} \) has to be calculated. In practice, however, since the NCS time delay typically consists of the forward channel delay, feedback channel delay, and controller processing delay, while the forward and feedback channels are symmetric, it is reasonable to assume that half of the overall delay comes from the forward channel while the other half is from the feedback channel. Hence, in the scheduling process, half of the delay \( T_{i1m} \) is assigned to the forward channel scheduler, and the other half is assigned to the feedback channel scheduler. In such a way, it is expected that the time slots reserved for the controller scheduler are enough for its scheduling operations. Hence, the longest allowable queuing delay \( T_{i1m} \) for those data packets could be determined as follows:

\[
T_{i1m} = \frac{1}{2} \left[ T_{i1M} - T_{i1}^{\text{tra}} (k - 1) - T_{i1}^{\text{pre}} (k - 1) - T_{i1}^{\text{pos}} (k - 1) \right],
\]

and the latest starting time would be

\[
\tau_{\text{ine}} = T_{i1m}.
\]

Moreover, since the noncoupled packets could be transmitted in an arbitrary order, the earliest transmission starting time of a packet could be set as \( \tau_{\text{inf}} = 0 \), while its execution time could be \( \tau_{\text{in}} = T_{i1}^{\text{tra}} \).

3.2.3. Scheduling for Hybrid Coupled and Noncoupled Information. In NCSs with both coupled and noncoupled information, assume that there are information coupled packets with \( u \) following \( [\tau_{\text{in}} \to \tau_{\text{in}}] (k = 1, \ldots, u) \) and \( v \) following \( [\tau_{\text{in}} \to \tau_{\text{in}}'] (h = 1, \ldots, v) \), which are determined by (11) and (13), respectively, as well as \( w \) information noncoupled packets. If there exists a parameter \( j (j \in (u - 2, u + w + 2)) \) for the scheduled packets \( \tau_{\text{in}}^{j} \) satisfying

\[
\begin{align*}
\tau_{\text{inf}} & \geq \tau_{\text{ina}} + \tau_{\text{inr}}^{j} \\
\tau_{\text{inf}} & \leq \tau_{\text{inf}} + \tau_{\text{inr}}^{j},
\end{align*}
\]

or
then we have the actual execution time for \( n \)th task of the \( i \)th plant as follows:

\[
\tau_{ina} = \max (\tau_{ina}^i, \tau_{ina}^j + \tau_{irr}^j).
\]

Hence, the time slot utilized for processing the packet within the control loop could be calculated as follows:

\[
\tau_{ina} + \tau_{ira}.
\]

The scheduling for any data packet to be transmitted within NCS could be conducted as shown in Algorithm 1.

3.3. Scheduling Mechanism for Controller Scheduler. To determine the scheduling operations for each packet, the execution time slot should be calculated for each data packet, and thus, the transmission time \( \tau_{irr} \), \( \tau_{ira} \), and \( \tau_{ina} \) should be calculated using (11) and (13) and (16)–(18), respectively. Finally, the transmission set \( \tau_{i} \) could be determined. The same scheduling operations shown in Algorithm 1 are executed within the controller scheduler for each packet of the plant.

While for any plant adopting an NCS structure as shown in Figure 1, the scheduling and transmission time slot distribution for each of its data packets could be illustrated in Figure 4.

4. Experimental Verification

To verify the effectiveness of the proposed time-delay bounded scheduling algorithm, experiments are conducted on a lab-customized teaching system. Figure 5 depicts the schematic of the system. As seen, it consists of four layers, with the first layer being the motion platforms connected to the local sensors and actuators and the second layer being the training platform server group connected with the campus network. The third layer is the school information center and the fourth layer is student clients, and they are connected to the public and the campus networks, respectively. Specifically, in our experiments, those motion platforms are located in the new campus acting as NCS control plants, and they are controlled by a local controller, while those student clients are located within the old campus acting as the remote controllers. As those two campuses are 34 kilometers away from each other, the teaching system within the two campuses is interconnected by public transmission networks. In the experiments, scheduling operations are conducted on the local controller and school information center server, respectively, to facilitate the packet transmissions.

The motion platform as shown in Figure 6 is a typical teaching instrument that has been commonly utilized in the laboratory for both load simulation and the position and attitude control of aircrafts. It has four degrees of freedom (4-DOFs), consisting of a linear motional freedom and three rational freedoms that are orthogonal to each other. Specifically, those 4-DOFs are along four separate axes, which are for inner ring motion, central ring motion, outer ring motion, and the line move motion, respectively, and each axis is equipped with two independent motors, with one for motion control and the other for force control. The force control system could be utilized to simulate the motion control loads or motion control interferences, while the motion control system could be used for force control interferences.

In our experiments, the 4-DOFs together with their respective two control systems are utilized to simulate eight independent control plants. Specifically, the system local controllers act as sensor schedulers for the forward channels to control all experimental components, and the main purpose of our experiment is to realize simultaneous closed-loop control of the 8 objects. To realize stable control of the control plants, we measured their dead zones and then set the measured values to be their respective lower and upper thresholds of the adopted relay-based dead-time compensators [29, 30] in the motion platform. In such a way, influences of the frictions within those objects could be eliminated. Each of those objects could be described with the mathematical model presented by (7), wherein discrete-time model matrices \( A \) and \( A_{id} \) denote the system state matrix and input matrix, respectively. Both \( A \) and \( A_{id} \) are shown in (20), and the system stability time-delay bound limits could be determined with the stability criterion presented in [27]. The model parameters, together with the loads of the four motion and the other four torque motors, i.e., inner ring, central ring, outer ring, and line move, are shown in Tables 1 and 2, respectively.

\[
A = \begin{bmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{bmatrix},
\]

\[
A_{id} = \begin{bmatrix}
A_{d11} & A_{d12} \\
A_{d21} & A_{d22}
\end{bmatrix}.
\]

To evaluate the effectiveness of the proposed time-delay bounded scheduling mechanism, we compare the time-delay performances of the tested motion platform with and without adopting the proposed scheduling mechanisms. Specifically, for an \( i \)th system, we define the control time of a control loop to be \( T_i(k) \), which starts from the \( k \)th control cycle to the time when the local controller sends its control signals. Table 3 presents the measured \( T_i(k) \) for the platform system without the proposed scheduling mechanism, while Table 4 shows the measured \( T_i(k) \) when the proposed scheduling mechanism is adopted. Results in Table 3 show that without the proposed scheduling mechanisms, there exist a number of measured delays, denoted with red color texts with \( T_i(k) > T_{i,\text{IM}} \), i.e., such delays are larger than the time-delay bound shown in Tables 1 and 2. Such large time delays could largely degrade the system performances or may even cause system instability. However, once the proposed data scheduling mechanisms are adopted, the measured time delays for all 8 channels are significantly reduced with \( T_i(k) < T_{i,\text{IM}} \). Such results indicate that with the
**Algorithm 1:** Scheduling mechanism in feedback channel for distributed NCSs.

**Input:** an existing transmission set $[\tau^{k}_{in} \rightarrow \tau^{j}_{jn}](k = 1, \ldots, u)$; a new data packet with $\tau_{in}$.

**Output:** a new transmission set $[\tau^{k}_{in} \rightarrow \tau^{j}_{jn}](k = 1, \ldots, u + 1)$.

1. Calculate the time-delay upper bound $T_{IM}$ with Theorem ?? for the data packet;
2. Calculate transmission priority $\tau_{in}$ with $T_{IM}$, i.e., $\tau_{in} \propto (1/T_{IM})$;
3. Compare calculated $\tau_{in}$ to $\tau_{in}$ of $\tau^{k}_{in}(k = 1, \ldots, u)$ in the set;
4. if calculated $\tau_{in} = \tau_{in}$ for $1 \leq k \leq u$ then
5. Insert $\tau_{in}$ into the end of set $[\tau^{k}_{in} \rightarrow \tau^{j}_{jn}](k = 1, \ldots, u)$, go to step 10;
6. else
7. Calculate information execution time $\tau_{ia}$ using equations (10)–(18);
8. Insert $\tau_{in}$ into set $[\tau^{k}_{in} \rightarrow \tau^{j}_{jn}](k = 1, \ldots, u)$ according to $\tau_{in}$ in an increasing order;
9. end if
10. Calculate the actual packet execution time $\tau_{ia}$ using equation (18);
11. Scheduling operation according to the order of transmission set $[\tau^{k}_{in} \rightarrow \tau^{j}_{jn}](k = 1, \ldots, u + 1)$;

**Figure 4:** The scheduling and execution time distribution for each packet of the plant adopting an NCS structure as shown in Figure 1.

**Figure 5:** Experimental setup of the NCS being utilized in our experiments.
time-delay bounds taken into account, the proposed scheduling mechanisms could help facilitate the data transmission for the plants, which, thus, helps realize stable system control.

The proposed scheduling mechanism together with different control schemes is also applied onto the teaching system to control the 4-DOFs motion platform online. Specifically, the control schemes are implemented to drive
the motion platform, and the proposed scheduling mechanism is utilized to facilitate the data transmission over the campus and the public communication networks. The dynamic performances of the teaching system with and without the scheduling mechanisms are evaluated in different cases.

(1) Scheduling effect verification with the classical PID control strategy: we compare the performances of the outer ring force system before and after the scheduling method being adopted. In the experiments, the influences of three main NCS delay factors, i.e., time delays, packet drops, and packet disordering, have been considered, wherein the influences of both packet drops and packet disordering were equivalently converted to be time delays. A simple PID controller was adopted for outer ring force system control, and the controller parameters were obtained with simple yet sophisticated modeling approach in the experiments [31], and they are also listed in Table 5. Once the controller parameters are obtained, they would remain fixed for the whole experiments. The step signals starting from $t = 0.1$ s are utilized as the system inputs; the theoretical and practical system responses are obtained. Figures 8 and 9 depict the theoretical and practical system responses with and without scheduling mechanisms being adopted, respectively. As seen in Figure 8, without adopting the scheduling mechanisms, the system response becomes divergent, i.e., the amplitude of the dashed curve increases with time and is away from the expected system response. In such a case, the system may finally become unstable. While once the scheduling mechanism is adopted, the system response converges, i.e., as shown in Figure 9, the practical system response converges to its expected state, and the system finally becomes stable. Such results demonstrate that the proposed scheduling algorithm could help facilitate the data transmission within NCS, which thus helps stabilize the system, even though the system dynamic performances are still not satisfactory as the system overshoot is large and the stability time is long as indicated in Figure 9.

(2) Scheduling verification with an NCS control mechanism proposed in [32]: we implemented the NCS process control mechanism presented in [32] and applied it to remote control the motion platform in our teaching system. Specifically, the proposed scheduling algorithm was utilized to facilitate data transmission within the NCS.

Figures 10 and 11 present the step responses of the outer ring force system without and with the proposed scheduling algorithm being adopted, respectively. Results demonstrate that, owing to its predictive ability, the implemented control mechanism helps achieve satisfactory control effects even without adopting the scheduling algorithm. However, once the proposed scheduling algorithm is further adopted, the system control effects could be largely improved. As can be seen in Figure 11, the overshoot amplitude, overshoot time,
and the stabilization time are significantly reduced as compared with those in Figure 10. Such performance comparison further verifies the effectiveness of the proposed scheduling algorithm.

As discussed, for an NCS system, the main factors affecting the system stability are the time delays, packet drops, and packet misorders of the transmitted data within the system; the main objective of NCS control system design is...
to achieve stable system control while minimizing the time delays. To address such an issue, the main idea of this study is to present a scheduling algorithm to reduce the time delays caused by data transmissions and postprocessing within NCS, i.e., $T_{1i}^{\text{wait}}(k)$, $T_{2i}^{\text{wait}}(k)$, $T_{1i}^{\text{pos}}(k)$, and $T_{2i}^{\text{pos}}(k)$. Meanwhile, with the system time-delay bounds of the closed-loop NCSs being taken into account, the data within the NCS could be transmitted more efficiently within the system stability delay bound. In such a way, the system stability could be guaranteed while the time delays could be largely reduced. The above experimental results convincingly demonstrated the effectiveness of the proposed time-delay-bounded scheduling algorithm for NCS. Due to the large amount of data to be scheduled in real-time data transmission process, however, the proposed scheduling algorithm still suffers from the heavy data processing load, especially when the real-time audio/video data is huge. Specifically, once the time delay exceeds the time-delay bound characterized by an existing system stability criterion, the system would still be instable.

5. Conclusion

This paper investigates the time-delay reduction issue in distributed NCSs with time-varying delays and presents a dual-way data scheduling algorithm with the system stability time-delay bounds taken into account. Specifically, the paper analyzes the influences of various time-delay variables on the system stability first and then presents a one-way scheduling mechanism for data transmission within NCS, followed by an extended dual-way scheduling algorithm for time-delay reductions in NCSs. To verify the effectiveness of the proposed scheduling algorithm, both simulations and experiments are finally conducted on a teaching system to remote control a motion platform in different cases. Results show that the proposed scheduling algorithm could largely facilitate data transmission over the networks, which thus helps improve the NCS stability and control accuracy in different cases. It is believed that with the system time-delay bounds being considered, such a proposed scheduling algorithm could not only help reduce time-delays in NCSs, but also achieve stable system control for NCSs.

Data Availability

All data, models, or code generated or used during the study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to acknowledge the financial support provided by the Fundamental Research Funds for the Central Universities (Grant no. G2018KY0308), China Postdoctoral Science Foundation (Grant no. 2018M641013), Postdoctoral Science Foundation of Shaanxi Province (Grant no. 2018BSHYDZZ05), and the Seed Foundation of Innovation and Creation for Graduate students in Northwestern Polytechnical University (Grant nos. ZZ2019028 and ZZ2019191).

References

[1] T. C. Yang, "Networked control system: a brief survey," *IEEE Proceedings—Control Theory and Applications*, vol. 153, no. 4, pp. 403–412, 2006.
[2] K.-Y. You and L.-H. Xie, "Survey of recent progress in networked control systems," *Acta Automatica Sinica*, vol. 39, no. 2, pp. 101–117, 2013.
[3] W.-A. Zhang and L. Yu, "Modelling and control of networked control systems with both network-induced delay and packet-dropout," *Automatica*, vol. 44, no. 12, pp. 3206–3210, 2008.
[4] R. A. Gupta and M.-Y. Chow, "Networked control system: overview and research trends," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 7, pp. 2527–2535, 2010.
[5] M. S. Mahmoud and M. M. Hamdan, "Fundamental issues in networked control systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 5, pp. 902–922, 2018.
[6] X.-M. Zhang, Q.-L. Han, X. Ge et al., "Networked control systems: a survey of trends and techniques," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 1–17, 2019.
[7] Y. Tipsuwan and M.-Y. Chow, "Control methodologies in networked control systems," *Control Engineering Practice*, vol. 11, no. 10, pp. 1099–1111, 2003.
[8] E. Joellaito, "Networked control systems: time delays and robust control design issues," in *Proceedings of the 2011 2nd International Conference on Instrumentation Control and Automation*, pp. 16–25, IEEE, Bandung, Indonesia, November 2011.
[9] X. Ge, F. Yang, and Q.-L. Han, "Distributed networked control systems: a brief overview," *Information Sciences*, vol. 380, pp. 117–131, 2017.
[10] H. Benítez-Pérez, J. L. Ortega-Arjona, P. E. Méndez-Monroy, E. Rubio-Acosta, and O. A. Esquivel-Flores, "Modelling of networked control systems," in *Control Strategies and Co-Design of Networked Control Systems*, pp. 25–40, Springer, Berlin, Germany, 2019.
[11] P. M. H. Heemels, A. R. Teel, N. Van de Wouw, and D. Nešić, "Networked control systems with communication constraints: tradeoffs between transmission intervals, delays and performance," *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1781–1796, 2010.
[12] D. Simon, Y.-Q. Song, and C. Aubrun, *Co-design Approaches to Dependable Networked Control Systems*, John Wiley & Sons, Hoboken, NJ, USA, 2013.
[13] D. Freirich and E. Fridman, "Decentralized networked control of systems with local networks: a time-delay approach," *Automatica*, vol. 69, pp. 201–209, 2016.
[14] K. Liu, A. Selivanov, and E. Fridman, "Survey on time-delay approach to networked control," *Annual Reviews in Control*, vol. 48, pp. 57–79, 2019.
[15] Y.-R. Zhao, G.-P. Liu, Y. Kang, and L. Yu, "Stability of networked control systems: a new time delay systems approach," in *Packet-Based Control for Networked Control Systems*, pp. 87–97, Springer, Berlin, Germany, 2018.
[16] X. Wang and M. D. Lemmon, "Event-triggered broadcasting across distributed networked control systems," in *Proceedings of the 2008 American Control Conference*, pp. 3139–3144, IEEE, Seattle, WA, USA, June 2008.
[17] X. Wang and M. D. Lemmon, “Decentralized event-triggered broadcasts over networked control systems,” in Hybrid Systems: Computation and Control, pp. 674–677, Springer, Berlin, Germany, 2008.

[18] X. Wang and M. D. Lemmon, “Event-triggering in distributed networked systems with data dropouts and delays,” in Hybrid Systems: Computation and Control, pp. 366–380, Springer, Berlin, Germany, 2009.

[19] Z. Gu, Z. Huan, D. Yue, and F. Yang, “Event-triggered dynamic output feedback control for networked control systems with probabilistic nonlinearities,” Information Sciences, vol. 457-458, pp. 99–112, 2018.

[20] Y.-L. Wang, C.-C. Lim, and P. Shi, “Adaptively adjusted event-triggering mechanism on fault detection for networked control systems,” IEEE Transactions on Cybernetics, vol. 47, no. 8, pp. 2299–2311, 2017.

[21] H. Sun, C. Peng, Y. Wang, and Y.-C. Tian, “Output-based resilient event-triggered control for networked control systems under denial of service attacks,” IET Control Theory & Applications, vol. 13, no. 16, pp. 2521–2528, 2019.

[22] C. Peng and F. Li, “A survey on recent advances in event-triggered communication and control,” Information Sciences, vol. 457-458, pp. 113–125, 2018.

[23] C. Ma, S. Chen, and W. Liu, “Maximum allowable delay bound of networked control systems with multi-step delay,” Simulation Modelling Practice and Theory, vol. 15, no. 5, pp. 513–520, 2007.

[24] R. Luck and A. Ray, “An observer-based compensator for distributed delays,” Automatica, vol. 26, no. 5, pp. 903–908, 1990.

[25] C.-H. Chen, C.-L. Lin, and T.-S. Hwang, “Stability of networked control systems with time-varying delays,” IEEE Communications Letters, vol. 11, no. 3, pp. 270–272, 2007.

[26] L. Zhang, Y. Shi, T. Chen, and B. Huang, “A new method for stabilization of networked control systems with random delays,” IEEE Transactions on Automatic Control, vol. 50, no. 8, pp. 1177–1181, 2005.

[27] A. Seuret, F. Gouaisbaut, and E. Fridman, “Stability of discrete-time systems with time-varying delays via a novel summation inequality,” IEEE Transactions on Automatic Control, vol. 60, no. 10, pp. 2740–2745, 2015.

[28] L. Zhang, H. Gao, and O. Kaynak, “Network-induced constraints in networked control systems—a survey,” IEEE Transactions on Industrial Informatics, vol. 9, no. 1, pp. 403–416, 2012.

[29] J. E. Normey-Rico and E. F. Camacho, “Dead-time compensators: a survey,” Control Engineering Practice, vol. 16, no. 4, pp. 407–428, 2008.

[30] K. K. Tan, T. H. Lee, and R. Ferdous, “Online relay automatic tuning of multi-loop pi controllers,” Intelligent Automation & Soft Computing, vol. 9, no. 3, pp. 155–167, 2003.

[31] J. B. Rawlings and D. Q. Mayne, Model Predictive Control: Theory and Design, Nob Hill Publishing, Madison, WI, USA, 2009.

[32] T. Wang, H. Gao, and J. Qiu, “A combined adaptive neural network and nonlinear model predictive control for multirate networked industrial process control,” IEEE Transactions on Neural Networks and Learning Systems, vol. 27, no. 2, pp. 416–425, 2016.