IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 21, 2022, accepted May 30, 2022, date of publication June 8, 2022, date of current version June 21, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3180735

Synthesis of a Two Degrees of Freedom Wireless Networked Digital Servo Control System With Dynamic State Prediction Based on a Gated Recurrent Unit-Based Round Trip Time Predictor

TATSUKI NONOMURA AND FUMITAKE FUJII, (Member, IEEE)
Department of Mechanical Engineering, Yamaguchi University, Yamaguchi 755-8611, Japan
Corresponding author: Fumitake Fujii (ffujii@yamaguchi-u.ac.jp)

ABSTRACT A wireless networked control system (WNCS) may suffer from an abrupt and large network communication delay that may degrade the control performance and lead to the instability of the closed-loop system. A two degrees of freedom (DOF) digital servo system for a WNCS is proposed in this paper. We trained a gated recurrent unit network to predict the round trip time (RTT) of a closed-loop packet with high accuracy and integrated it with a feedback controller connected to the network. The feedback controller was configured to monitor the behavior of RTTs for each node in the network, predict the future behavior of the nodes, and synthesize a control input that compensates for the worst case RTT of the nodes using the dynamic state predictive control scheme. We constructed a pendulum manipulator system, in which an inverted pendulum is attached at the tip of the six DOF industrial manipulator. The control objective of a pendulum manipulator system was to track the first axis angle of the manipulator to the given reference trajectory while stabilizing the pendulum around its unstable equilibrium. The results of the numerical simulations and experiments were reported to validate the performance of the proposed controller in a WNCS that occasionally suffers from a large network transmission delay that amounted to several tens of sampling intervals at the worst case.

INDEX TERMS Wireless networked control system, dynamic state predictive control, the Internet of Things, gated recurrent unit network.

I. INTRODUCTION

With the advancement in broadband telecommunication network infrastructure, the Internet of Things technology has been deployed in industries and consumer markets. Many electrical appliances are now able to connect to the internet. A networked control system (NCS) is a feedback control system, in which the plant is controlled using network data transmission. Examples of NCSs include remote-controlled cars and drones, rescue robots, and telesurgery systems.

A recent trend in the development of NCSs in industry is the utilization of wireless network communication. Although wired network connections are firmer and more robust than wireless network connections, network cables can be voluminous and expensive for large-scale networks. Physical cables can restrict the motion of mobile robots and/or rotating machines. Replacing such cable connections with wireless network devices can significantly reduce the initial installation cost and increase the degrees of freedom (DOF) in the deployed NCS.

Issues such as packet loss, quantization error, transmission delay, and degradation of the control performance metrics are encountered with respect to feedback control over the network. Transmission delay is a critical issue in remote control systems, as the feedback control system may lose stability when it suffers from a large signal transmission delay [1]. This problem worsens if the signal transmission is implemented over a wireless network, as its transmission delay increases significantly when network congestion occurs [2]. Therefore, in this study, we focused on compensating for the delay in wireless NCSs (WNCSs).

Several methods have been proposed to compensate for the delays in feedback control systems. These methods include the Smith predictor [3], internal model control [4], and state

The associate editor coordinating the review of this manuscript and approving it for publication was Azwirman Gusrialdi.

predictive control [5]. Although these methods work well for constant delays, their performance deteriorates when the delay varies, which occurs in NCSs. Several studies have been conducted to compensate for the time-varying delay in NCSs [6]–[13]. It was shown in [6]–[9] that linear matrix inequality (LMI)-based methods are effective in compensating for the time-varying delay, provided that its variation is small. However, it might be difficult to apply the results directly to WNCSs as a large transmission lag can occasionally occur.

Yoshida *et al.* [12] proposed a dynamic state predictive control for stabilizing an inverted pendulum. They calculated state predictions using the measured round trip time (RTT) of the target network and synthesized control inputs with a feedback gain determined by the optimal control theory. They performed 100 numerical simulations with three different measured RTT sequences and showed that stability was maintained under RTT variations of 200 ms with a 90% probability. However, as the RTT is an unknown quantity, the closed-loop stability can be improved if the RTT uncertainty in the synthesis of feedback controllers is considered. Liu *et al.* [13] proposed a packet-based dynamic state predictive control based on time synchronization between the controller and actuator nodes. However, this method might further increase the RTT when the network encounters a large transmission delay. Moreover, performing time synchronization in a large scale NCS that includes many network nodes is significantly difficult.

The RTT in a network communication varies depending on the performance of the network device, the physical distance between network devices [14], the time of day when network communication occurs [15], and the sampling interval [16]. Depending on the size of the buffer, the RTT can be as high as 1 s if the network is congested [17]. However, if a high precision RTT predictor is synthesized, time synchronization is not required. The development of RTT predictor models has been reported in the literature. Two different approaches exist: one approach includes the application of recurrent neural networks (RNNs) [15], [18]–[20] and the other approach utilizes probabilistic modeling [21], [22]. In the latter approach [22], the behavior of the RTT was modeled using a Markov process, whose transition probability was determined based on beta distribution functions. In addition, a switching algorithm for the stabilizing feedback control gain was proposed based on the estimated interval of the transient changes of the RTT.

We recently reported the synthesis of a two DOF digital servo NCS using an RNN-based RTT predictor [23] for a virtual inverted pendulum that was controlled over a network. We verified the performance of our two DOF WNCS using a hardware-in-the-loop setup and succeeded in stabilizing the virtual pendulum while tracking the step changes in the first joint axis angle of the manipulator.

In this study, we addressed the control of a real pendulum manipulator system over a wireless network for the stabilization of the pendulum attached at the tip of a six DOF industrial manipulator while performing servo control of the first joint axis of the manipulator. The angles of the first joint of the manipulator and the inverted pendulum were measured independently to form two isolated sensor nodes in the WNCS. We also improved the RTT predictor for dynamic state predictive control. We employed a gated recurrent unit (GRU) network and trained it to predict several future RTT samples to compensate for the phase lag introduced by a Kalman filter, which is used to smooth unnecessary jagged changes in the predicted RTT. We performed both numerical simulations and real experiments to demonstrate that the proposed dynamic state predictive control exhibits better performance than the previous dynamic state predictive control methods.

The remainder of this paper is organized as follows. In Section II, the measurement of the end-to-end (E2E) delay and the RTT in our WNCS are provided in detail and the problem formulation for the synthesis of the two DOF digital servo controller over a wireless network is described. In Section III, the conventional two DOF feedback controller design proposed in the literature, which was used to determine the feedback control structure, is introduced. Then, a gain synthesis process that guarantees closed-loop stability when it is implemented over a wireless network is described. We also explain the Kalman filter that was used for state estimation. In Section IV, we propose a dynamic state predictive two DOF servo controller for our pendulum manipulator system, whose prediction horizon is determined by the output of the GRU-based RTT predictor. In Section V, the performance of the proposed control system is verified through numerical simulations. The results of real experiments are reported in Section VI. The conclusions and future work implications for further performance improvements are presented in Section VII.

II. PROBLEM FORMULATION

We first investigate the delays observed in wireless network communication and mathematically formulate the controller synthesis problem for a WNCS. There are four primary sources of network transmission delay: serialization delay D_s, propagation delay D_{prop}, processing delay D_{proc}, and queuing delay D_q [24]. Both D_s and D_{prop} depend on the packet size and the physical distance between two network nodes and are assumed to be constant. Conversely, $D_{proc}(t)$ and $D_q(t)$ vary depending on the traffic of the network. We measured the E2E delay, which is the summation of the four delays, and the RTT of a packet originating from a node of the network.

A. IDENTIFICATION OF THE END-TO-END DELAY AND ROUND TRIP TIME (RTT)

To understand how the RTT changes in a network communication with wireless nodes, we set up an experimental environment, as illustrated in Fig. 1. We identified the E2E delay and RTT of the network. Two Raspberry Pi computers (Raspberry Pi 3 Model B+, Raspberry Pi Foundation) and a WiFi router (WSR-1166DHPL2/N, Buffalo Inc., Japan) were

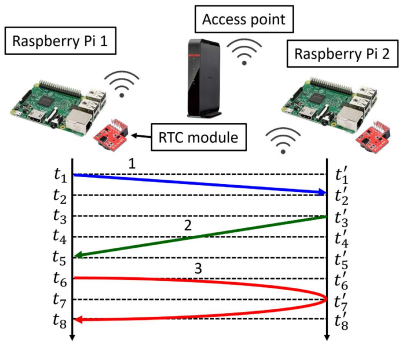


FIGURE 1. **Measurement of the end-to-end (E2E) delay and round trip time (RTT).**

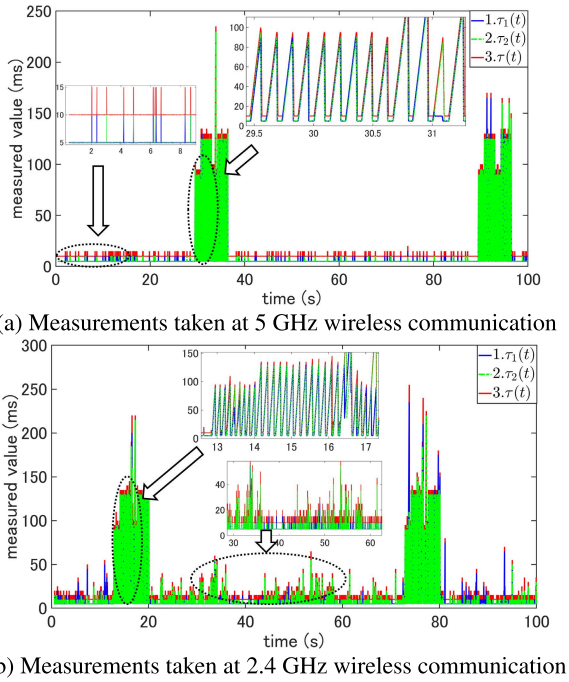
used. The router was connected to the university network as an access point. We tested the IEEE 802.11ac and 802.11n standards for the wireless connection with 5 and 2.4 GHz frequency bands, respectively. Because the Raspberry Pi is not equipped with a real time clock (RTC) module, we added one (Raspberry Pi RTC Expansion Module v1.1, Seeed Studio, China) for the experiment.

We connected the two general-purpose input/output (GPIO) pins of each Raspberry Pi board with a wire. We modified the pin voltage level from low to high on one board, hereafter referred to as the leader. The other board, hereafter referred to as the follower, was programmed to continuously monitor the pin status, and start the remaining part of the program immediately after a change in the status of the GPIO pin was sensed. The RTCs of the two Raspberry Pi boards were synchronized accordingly.

After the completion of time synchronization, both boards were programmed to repeatedly perform socket communications at 5 ms intervals. The user datagram protocol was used for packet communication. The leader transmitted a 12-byte packet containing the current RTC readout and the serial packet counter that was sequentially incremented on the leader board. The follower recorded the RTC value when it received a packet from the leader and then generated a 24-byte packet, which included the 12-byte packet received from the leader, the RTC readout of the follower when the follower received the packet, and a serial counter that was generated inside the follower board.

If the leader or follower did not receive a packet in the current sampling interval, we used the preceding packet to calculate the E2E delay and RTT. The following three quantities were determined for each sampling interval:

1) $\tau_1(t)$: The E2E delay identified by the difference between the time when the follower received a packet from the leader and the leader RTC included in the received packet.
2) $\tau_2(t)$: The E2E delay identified by the difference between the time when the leader received a packet from the follower and the follower RTC included in the received packet.
3) $\tau(t)$: The RTT as identified by the difference between the time when the leader transmitted a packet and the time when the leader received a returning packet from the follower.



(a) Measurements taken at 5 GHz wireless communication

(b) Measurements taken at 2.4 GHz wireless communication

FIGURE 2. **Measurements of the E2E delay and RTT.**

The results are shown in Fig. 2. The plots shown in Figs. 2a and 2b exhibit a large transmission delay for a duration of 6–7 s in both frequency bands. During the time window of congestion, the two E2E delays τ_1 and τ_2 and the RTT τ were identified to be approximately the same. This result implies that a communication delay occurs in both directions of packet transmission when the network is congested. A packet was maintained in transit between the two nodes during the congestion period of 100–150 ms, and its returning packet was delivered immediately after the delivery of the delayed packet. We did not observe a significant difference between the largest delays in the two frequency bands; however, the perturbations of delay were observed more frequently in the 2.4 GHz wireless communication than in the 5 GHz wireless communication. We also noticed the existence of protuberances of the RTTs over the entire measurement range in both frequency bands. Since we quantify the RTTs as the integer multiples of the sampling interval, a small RTT variation would result in small protuberances of five ms, as can be clearly seen in the zoomed-in portion of the first eight seconds of the measurement in Fig. 2a. Another possible cause of the observed protuberances is the inaccuracy of the real time clock module.

Based on the behavioral knowledge of network transmission delay, we mathematically formulated the wireless networked control problem.

B. MATHEMATICAL FORMULATION OF THE WIRELESS NETWORKED CONTROL PROBLEM

Fig. 3 shows the block diagram of the WNCS studied in this paper. This system is composed of a continuous-time

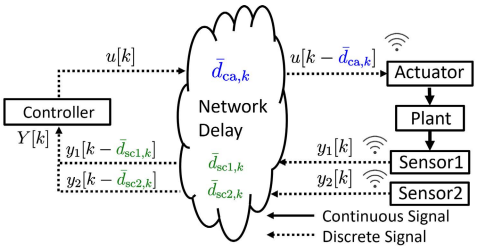


FIGURE 3. Block diagram of a wireless networked control system (WNCS).

plant, a discrete time controller, and two wireless sensors labeled Sensor1 and Sensor2, which are used to measure two independent output quantities. The plant is classified as a linear time-invariant single-input multiple-output system, whose state space model is given by

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t), \end{aligned} \tag{1}$$

where $x(t) \in \mathbb{R}^n$, $y(t) \in \mathbb{R}^l$, and $u(t) \in \mathbb{R}$ represent the state, output, and control inputs, respectively. Matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times 1}$, and $C \in \mathbb{R}^{l\times n}$ define the realization of the plant. We assume that the pair (A, B) is stabilizable and (C, A) is detectable. We also assume that the loop transmission delay of signals at the k-th sampling interval τ_k is quantified as an integer multiple of the sampling interval h, which is denoted by

$$\tau_k = d_k h \quad (d_k \in \mathbb{N}). \tag{2}$$

We assume that the multiple d_k takes a bounded value and belongs to the set $\Omega \triangleq \{d_m, \cdots, d_M\}$. However, the boundaries d_m and d_M are assumed to be unknown. Using this notation, we hereafter denote the transmission delay from the controller to the actuator as $\bar{d}_{\mathrm{ca},k}$, that from Sensor1 to the controller as $\bar{d}_{\mathrm{sc1},k}$, and that from Sensor2 to the controller as $\bar{d}_{\mathrm{sc2},k}$. We also assume that both the controller and actuator nodes use the latest delivered packet and use the previously delivered packet when no packet is delivered to the nodes in the current sampling interval. Thus, $\bar{d}_{*,k}$, where $*$ represents ca, sc1 or sc2, in Fig. 3 is defined as

$$\bar{d}_{*,k} = \begin{cases} d_{*,(k-1)} + 1, & \text{no packet received,} \\ d_{*,k}, & \text{packet received.} \end{cases} \tag{3}$$

We can discretize the plant dynamics of (1) by

$$\begin{aligned} x[k+1] &= A_d x[k] + B_d u[k - \bar{d}_{\mathrm{ca},k}] \\ y[k] &= Cx[k] \end{aligned} \tag{4}$$

using a zero-order holder, where A_d and B_d are given by

$$A_d = e^{Ah}, \quad B_d = \int_0^h e^{At} B dt, \tag{5}$$

respectively. Considering the transmission delay from the two sensor nodes to the controller, the output $Y[k]$ observed by the controller node is described as

$$Y[k] = \begin{bmatrix} y_1[k - \bar{d}_{\mathrm{sc1},k}] \\ y_2[k - \bar{d}_{\mathrm{sc2},k}] \end{bmatrix} = \begin{bmatrix} C_1 x[k - \bar{d}_{\mathrm{sc1},k}] \\ C_2 x[k - \bar{d}_{\mathrm{sc2},k}] \end{bmatrix}. \tag{6}$$

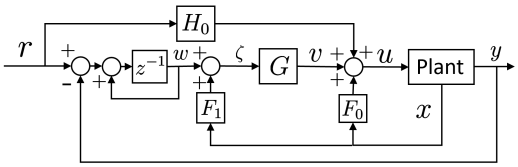


FIGURE 4. Two degrees of freedom (DOF) digital servo system.

III. EXTENSION OF A TWO DEGREES OF FREEDOM (TWO DOF) DIGITAL OPTIMAL SERVO CONTROL SYSTEM TO A WIRELESS NETWORKED CONTROL SYSTEM (NCS)

We aim to synthesize a two DOF digital servo control system that maintains its function even during the occurrence of occasional large transmission delays. To achieve this, we applied the feedback structure of the two DOF digital optimal servo controller proposed in the literature and synthesized the included gains so that the closed-loop system maintains its stability under a large signal transmission delay.

A. SYNTHESIS OF THE TWO DOF DIGITAL OPTIMAL SERVO CONTROLLER

Fig. 4 shows the feedback structure of the digital servo controller proposed by Hagiwara *et al.* [25]. As it utilizes a two DOF structure, we could independently design reference tracking and disturbance rejection properties. Using the symbols introduced in Fig. 4, the control law is given by

$$u[k] = F_0 x[k] + H_0 r + v[k]. \tag{7}$$

The gains F_0 and G were determined to minimize the quadratic performance indices J_1 and J_2, which are defined as follows:

$$J_1 = \sum_{k=0}^{\infty} \left\{ \tilde{x}[k]^T Q \tilde{x}[k] + \tilde{u}[k]^T R \tilde{u}[k] \right\} \tag{8}$$

and

$$J_2 = \sum_{k=0}^{\infty} \left\{ \tilde{\zeta}[k]^T \Omega \tilde{\zeta}[k] + \tilde{v}[k]^T \Theta \tilde{v}[k] \right\}, \tag{9}$$

where Q, R, Ω, and Θ are selected to be positive definite. The quantities $\tilde{x}[k] = x[k] - x_\infty$, $\tilde{u}[k] = u[k] - u_\infty$, $\tilde{\zeta}[k] = \zeta[k] - \zeta_\infty$, and $\tilde{v}[k] = v[k] - v_\infty$ that appear in the equations for J_1 and J_2 represent the difference between the current values and the steady-state values.

The feedforward gain H_0 should be determined to equate the output y to its reference r in the steady state. The steady-state calculation yields H_0, which is determined as follows:

$$H_0 = -[C(A_d + B_d F_0 - I)^{-1} B_d]^{-1}. \tag{10}$$

The feedback input signal $v[k]$ takes a nonzero value only when the output deviates from the reference, which is caused by modeling inaccuracy and/or disturbance. The gain F_1 is formulated as follows:

$$F_1 = C(A_d + B_d F_0 - I)^{-1}. \tag{11}$$

B. DETERMINATION OF F_0 AND G IN THE PRESENCE OF A FEEDBACK TRANSMISSION DELAY

We then extended the two DOF controller design to our WNCS. Let us start from the simplest case by assuming that all state variables can be measured directly by the installed wireless sensors. If we additionally assume that all sensor-to-controller delays are identical, applying the control law (7) results in closed-loop dynamics of the NCS that is algebraically characterized by

$$\bar{x}[k+1] = \bar{A}\bar{x}[k] + \bar{B}\bar{x}[k-\bar{d}_k], \tag{12}$$

where $\bar{x}[k] = [x[k], \zeta[k]]^T$. The closed-loop matrices are obtained as follows:

$$\bar{A} = \begin{bmatrix} A_d & 0 \\ -F_1B_dF_0 & 1 \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} B_dF_0 & B_dG \\ F_1B_dF_0 & F_1B_dG \end{bmatrix}, \tag{13}$$

where $\bar{d}_k = \bar{d}_{\mathrm{ca},k} + \bar{d}_{\mathrm{sc1},k}$. As observed in the previous section, $\bar{d}_k$ can be significantly perturbed. As a result, the feedback control system may lose its stability if we use the nominal gains that minimize (8) and (9). Therefore, we must determine the appropriate gains while considering the delay $\bar{d}_k$.

Zhao *et al.* [8] proposed the synthesis of a state feedback control law $u[k] = F_0x[k]$ that stabilizes the closed-loop system as follows:

$$x[k+1] = A_dx[k] + B_dF_0x[k-\bar{d}_k]. \tag{14}$$

They designed a specific Lyapunov function that includes the past states determined by the smallest and largest possible delays d_m and d_M, respectively, and derived the corresponding LMI conditions by applying the Schur complement. We followed their procedure to determine the state feedback gain matrix F_0. We should note here that the assumption of identical sensor-to-controller delays is somewhat invalid. As shown in the experimental validations later in this paper, the two sensor-to-controller delays frequently exhibited different values. However, this assumption is required for the analytical treatment of the synthesis of F_0.

Once F_0 is determined, the overall closed-loop stability depends on the choice of G. We first synthesized F_0 for the worst case delay $\bar{d}_k = d_M$ and calculated F_1 using (11). We found that the closed-loop dynamics of our WNCS corresponding to the worst case delay ($\bar{d}_k = d_M$) were characterized by

$$\xi[k+1] = \Phi\xi[k], \tag{15}$$

where the closed-loop matrix Φ is given by

$$\Phi = \begin{bmatrix} A_d & 0 & \cdots & 0 & B_dF_0 & 0 & \cdots & 0 & B_dG \\ I & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 & 0 & \cdots & 0 & 0 \\ -F_1B_dF_0 & 0 & \cdots & 0 & F_1B_dF_0 & I & \cdots & 0 & F_1B_dG \\ 0 & 0 & \cdots & 0 & 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & I & 0 \end{bmatrix}$$

and $\xi[k]$ is its state vector, which is defined as

$$\xi[k] = \Big[x^T[k], x^T[k-1], \cdots, x^T[k-d_M], \zeta^T[k], \zeta^T[k-1], \cdots, \zeta^T[k-d_M]\Big]^T.$$

We manually determined G such that all eigenvalues of Φ were located inside the unit circle of the complex plane.

C. STATE ESTIMATION USING A STEADY-STATE KALMAN FILTER

Although we assumed that all state vector components were directly measurable in our previous development, this is not always the case. Consequently, we synthesized a steady-state Kalman filter to obtain state vector estimates. Let $\hat{x}[k]$ be the estimate of $x[k]$. Then, the Kalman filter is formulated as follows:

$$\hat{x}[k+1] = A_d\hat{x}[k] + B_du[k] + K(Y[k] - C\hat{x}[k]); \tag{16}$$

Moreover, the control input $u[k]$ is synthesized as follows:

$$u[k] = F_0\hat{x}[k] + H_0r + v[k]. \tag{17}$$

IV. THE IMPLEMENTATION OF A TWO DOF SERVO SYSTEM WITH DYNAMIC STATE PREDICTIVE CONTROL USING A GATED RECURRENT UNIT (GRU)-BASED RTT PREDICTOR

We propose the integration of a dynamic state predictor into the synthesized WNCS in this section. We also propose the repeated use of a pretrained GRU network that is used to predict the next step RTT to calculate the several steps ahead RTT of the network to determine the prediction horizon in dynamic state prediction. Based on the inferences deduced from the RTT measurements in Section II, we have to assume the occurrence of a large RTT that occasionally exceeds the largest d_M which was set during the synthesis of F_0. The introduction of the dynamic state predictive control scheme in our WNCS is expected to compensate for a significantly large transmission delay caused by the congestion of the network and improve the stability margin of the closed-loop WNCS, provided that future RTT values are predicted with high accuracy. For this purpose, we monitored the behavior of the RTT in real time during the control operation.

A. CALCULATION OF THE DYNAMIC STATE PREDICTION USING THE GRU-BASED RTT PREDICTOR

The RTT of a wireless network varies significantly, as illustrated in Fig. 2, and we cannot predict when the network will be in a state of congestion. Therefore, we aimed to train a GRU model using the measured RTT sequence of the target network to predict future RTT behavior, including possible congestion. We recently reported a hardware-in-the-loop simulation result of the dynamic state predictive control for a pendulum manipulator system [23], in which we used a conventional RNN to predict a one sampling step ahead RTT. The proposed control law is given as follows:

$$u[k] = F_0\hat{x}[k+H_k] + H_0r + v[k]. \tag{18}$$

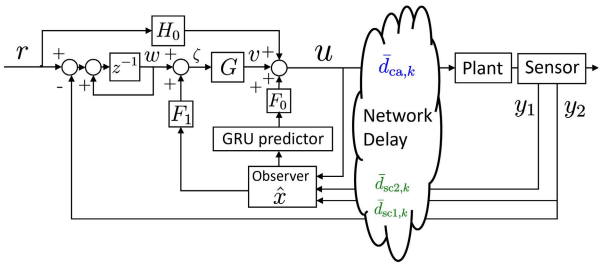


FIGURE 5. Proposed two DOF digital servo WNCS with a switching-type dynamic state predictive controller.

The dynamic state prediction $\hat{x}[k+H_k]$ is calculated as follows:

$$\hat{x}[k+H_k] = A_d^{H_k}\hat{x}[k] + \sum_{j=k-H_k}^{k-1} A_d^{k-j-1}B_d u[j], \tag{19}$$

where H_k denotes a one step ahead RTT generated by the RNN-based RTT predictor at time step k. Because there was only one physical sensor in our previous experimental target, we concentrated on the prediction of one sensor-to-controller delay of a WNCS. However, in the case of a real pendulum manipulator system, two different RTTs must be considered for two sensor nodes, i.e., Sensor1 and Sensor2 in this study, as illustrated in Fig. 3.

Therefore, we trained two GRUs to predict the two RTTs, namely, the RTT for the controller and Sensor1 network and that for the controller and Sensor2 network. As the actuator is controlled by a notebook PC that also retrieves the Sensor1 output in our pendulum manipulator system, we only need to predict the two aforementioned RTTs. Let $\hat{H}_{1,k}$ and $\hat{H}_{2,k}$ be the predicted RTTs for the controller-Sensor1 and controller-Sensor2 networks, respectively. Next, we propose the following switching control law for our WNCS.

$$u[k] = \begin{cases} F_0\hat{x}[k+\hat{H}_{1,k}] + H_0 r + v[k] & \text{if } \hat{H}_{1,k} > \hat{H}_{2,k} \\ F_0\hat{x}[k+\hat{H}_{2,k}] + H_0 r + v[k] & \text{if } \hat{H}_{2,k} > \hat{H}_{1,k} \end{cases} \tag{20}$$

The predicted RTT with the greatest value will be recognized as the prediction horizon of the dynamic state prediction to compensate for the worst case transmission delay.

B. CONFIGURING THE GRU FOR RTT PREDICTION

In this subsection, the calculations of the RTT predictions $\hat{H}_{*,k}(* = 1, 2)$ are described. Our previous study [23] revealed that both the measured and the predicted RTTs may exhibit large and rapid changes. As the straightforward use of the raw predicted RTT sequence in the dynamic state predictive control might deteriorate the control performance, we smoothed the predicted RTT sequences by synthesizing another Kalman filter. Furthermore, we decided to predict $H(>1)$ steps ahead RTTs of the two controller-sensor network connections and applied them to the synthesized Kalman filter to mitigate the phase lag introduced by filtering.

The relationship between the input and output quantities of the GRU network in this study is specified by the following set of equations:

$$r[k] = \sigma(W_r R[k] + U_r h[k-1] + b_r) \tag{21}$$
$$z[k] = \sigma(W_z R[k] + U_z h[k-1] + b_z) \tag{22}$$
$$\hat{h}[k] = \tanh\left(W_h R[k] + U_h(r[k] \odot h[k-1]) + b_h\right) \tag{23}$$
$$h[k] = z[k] \odot h[k-1] + (1 - z[k]) \odot \hat{h}[k] \tag{24}$$
$$y_{\mathrm{GRU}}[k] = Vh[k] + c, \tag{25}$$

where $R[k] = [\mathrm{RTT}[k-1], \mathrm{RTT}[k-2], \cdots, \mathrm{RTT}[k-5]]^T$ is an input vector for the predictor network and W_*, U_*, b_*, V, c represent the weights and biases of each layer and gate.

The network was configured to contain one hidden layer with 120 neurons. We used 50 previous RTT measurements $\{R[k-1], R[k-2], \cdots, R[k-\tau]\}$ $(\tau = 50)$ at time k to calculate the current hidden layer output $h[k]$. We used the previous five samples of RTT measurements $R[k]$ as the input of the GRU predictor to detect the network congestion that lasts for 6–7 s as early as possible. The networks were trained as follows. We prepared 500,000 RTT measurements of the two "controller and sensor" networks. We used 80% of the measurements for training and the remaining 20% for validation. We set the number of epochs to 10,000 and the batch size n to 1,024. The parameters were updated using AMSGrad [26], whose learning rate was set to 0.001. The parameters of the network W_*, U_*, V and b_*, c were updated to minimize the mean squared error, which is defined as

$$E = \frac{1}{n}\sum_{k=1}^{n}(y_{\mathrm{GRU}}[k] - \mathrm{RTT}[k])^2, \tag{26}$$

where $y_{\mathrm{GRU}}[k]$ is the output of the network and $\mathrm{RTT}[k]$ denotes the current measured RTT.

After the training was completed, we integrated the trained GRU models into the proposed control system to calculate the H steps ahead prediction of the RTT in the following manner. We stored the past-to-present $\tau + 4$ RTT measurements in a buffer to calculate $y_{\mathrm{GRU}}[k]$, which was used to form a new input layer $\hat{R}[k] = [y_{\mathrm{GRU}}[k], \mathrm{RTT}[k-1], \cdots, \mathrm{RTT}[k-4]]^T$ and calculate $y_{\mathrm{GRU}}[k+1]$ using (21)-(25). We repeated this procedure H times to determine the H steps ahead prediction of the RTT, i.e., $y_{\mathrm{GRU}}[k+H]$.

We finally applied an alternate Kalman filter, which is defined by

$$K[k] = \frac{(p[k-1]+r)}{p[k-1]+r+q} \tag{27}$$
$$\begin{aligned}\hat{y}_{\mathrm{GRU}}[k+H] = {} & \hat{y}_{\mathrm{GRU}}[k+(H-1)] \\ & + K[k](y_{\mathrm{GRU}}[k+H] \\ & - \hat{y}_{\mathrm{GRU}}[k+(H-1)])\end{aligned} \tag{28}$$
$$p[k] = (1 - K[k])(p[k-1]+r) \tag{29}$$
$$\hat{H}_k = \mathrm{Round}\left(\frac{\hat{y}_{\mathrm{GRU}}[k+H]}{h}\right) \tag{30}$$

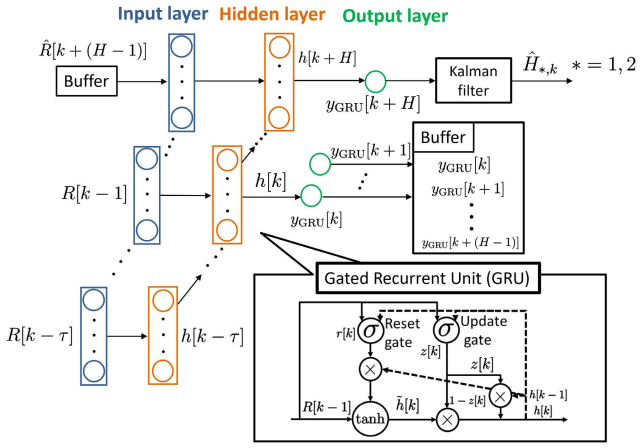


FIGURE 6. Structure of the gated recurrent unit-based RTT predictor.

to the sequence of $y_{\mathrm{GRU}}[k+H]$ to achieve the smoothed estimate $\hat{y}_{\mathrm{GRU}}[k+H]$, where r and q are the covariances of the measurement and process noises, respectively. An integer multiple expression $\hat{H}_k$ of $\hat{y}_{\mathrm{GRU}}[k+H]$ was needed, as defined by (30), to calculate the dynamic state prediction.

V. NUMERICAL SIMULATION OF THE WNCS

The details of the pendulum manipulator system are presented in this section and are followed by a description of the implementation of the proposed controller to our WNCS. Simulations of the proposed control system, in which physically moving bodies of the system and the sensor measurements are replaced with corresponding mathematical equations, were performed using two additional reference controllers to highlight the performance of the proposed control system.

A. EXPERIMENTAL APPARATUS AND CONTROL OBJECTIVES

Fig. 7 shows the pendulum manipulator system used in this study. The manipulator (MZ07-01, Nachi-Fujikoshi Corp., Japan) is a six DOF industrial manipulator. A pendulum is attached at the tip of the manipulator so that it can rotate freely around the axis. The control objective is to track the designated reference angle trajectory of the first joint axis of the manipulator while stabilizing the inverted pendulum in its upright position. We only used the first joint motor of the manipulator for this purpose and fixed the remaining five joints in the modeling and experimental runs.

The manipulator was controlled by its in-house controller (CFD, Nachi-Fujikoshi Corp., Japan). The in-house controller has a supervised control mode in which it receives joint angle references from the upper level controller, controls the manipulator based on the implemented control law and returns the result of control to the upper level controller. The commands and the results of control are transmitted over a TCP/IP network communication protocol between the upper level controller PC and the in-house controller.

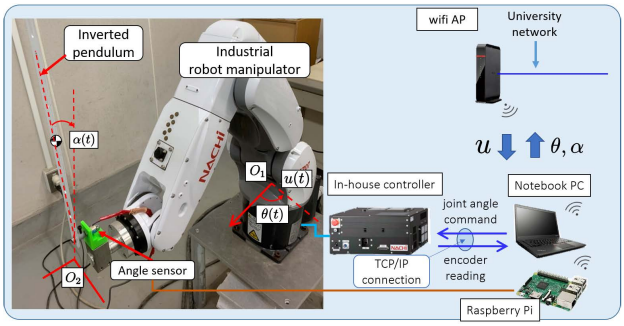


FIGURE 7. Pendulum manipulator system and its related physical quantities.

We empirically know that there exists a constant transmission lag in the network data transmission in this TCP/IP network communication. We assume that the delay amounts to five sampling steps, which is twenty-five ms.

As depicted in Fig. 7, the upper level controller (the notebook PC) of the manipulator also serves as the sensor node of the joint angle $\theta(t)$ of the manipulator. The angle of inverted pendulum $\alpha(t)$ was measured by a potentiometer (CP-2UN, Midori Precision, Japan) whose output was captured by the A/D converter (MCP3208, Microchip Technology, USA) connected to the Raspberry Pi 3B+, which served as another sensor node. These two sensor readings were fed back to the controller PC via the wireless network connection through the WiFi access point, as depicted in the figure.

As we concentrate on the numerical simulation in this section, the manipulator, its in-house controller and the potentiometer are replaced with a mathematical model of the system, which is derived as follows.

B. MATHEMATICAL MODELING OF THE PENDULUM MANIPULATOR SYSTEM AND CALCULATION OF CONTROLLER GAINS

We modeled the behavior of the pendulum manipulator system by assuming that the pendulum motion does not affect the first joint axis motion of the manipulator, as the pendulum is sufficiently lighter than the payload limit of the manipulator. The physical parameters that appear in the mathematical model of the system are listed in Table 1. We first modeled the dynamics in the continuous-time domain and discretized it for the successive synthesis of the proposed two DOF dynamic state predictive wireless servo NCS. Let $u(t)$ denote the input angle command of the first joint motor of the manipulator.

Because the behavior of each joint of a typical industrial manipulator is known to be approximated with a second-order model with sufficient accuracy [27], we assumed the following second-order linear differential equation:

$$\ddot{\theta}(t) + a_1\dot{\theta}(t) + a_2\theta(t) = b_1 u(t-L) \tag{31}$$

as the mathematical model of the first joint motion of the manipulator, where L is a known constant delay due to the TCP/IP network communication between the upper level

controller and the in-house controller of the manipulator. The parameters a_1, a_2, and b_1 were identified experimentally using the input and output sequences.

The behavior of the pendulum manipulator system was determined according to

$$\dot{x}(t) = Ax(t) + Bu(t - L), \tag{32}$$

where $x(t) = [\alpha(t), \dot{\alpha}(t), \theta(t), \dot{\theta}(t)]^T$ denotes the state vector. The matrices A and B are defined as follows:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \dfrac{m_1 g l_1}{\beta} & -\dfrac{D}{\beta} & \dfrac{m_1 L_0 l_1 a_2}{\beta} & \dfrac{m_1 L_0 l_1 a_1}{\beta} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -a_2 & -a_1 \end{bmatrix} \tag{33}$$

$$B = \left[0, -\frac{m_1 L_0 l_1 b_1}{\beta}, 0, b_1\right]^T \quad (\beta = J_1 + m_1 l_1^2). \tag{34}$$

By introducing the values of the parameters listed in Table 1 and discretizing (32) with a sampling interval $h = 5$ ms, we obtain

$$x[k+1] = \begin{bmatrix} 1.0003 & 0.0050 & 0.0091 & 0.0009 \\ 0.1238 & 1.0003 & 3.5527 & 0.3547 \\ 0 & 0 & 0.9957 & 0.0046 \\ 0 & 0 & -1.6559 & 0.8347 \end{bmatrix} x[k] + \begin{bmatrix} -0.0091 \\ -3.5527 \\ 0.0043 \\ 1.6559 \end{bmatrix} u[k-5]. \tag{35}$$

Next, we calculated the gain matrices of the proposed controller. The gain F_0 was determined by solving the LMIs, as explained in Section III, using the SeDuMi [28] solver in MATLAB. The maximum theoretical tolerance of the transmission delay was determined to be $d_M = 17$. The associated feedback gain G was determined manually such that the matrix Φ in (15) had all its eigenvalues inside the unit circle. Because direct measurements of the angular velocities $\dot{\alpha}(t)$ and $\dot{\theta}(t)$ were unavailable, we synthesized a Kalman filter to estimate the state vector. The controller gains are subsequently summarized.

$$F_0 = \begin{bmatrix} 1.9624 & 0.3927 & 2.4459 & 0.7679 \end{bmatrix}$$
$$F_1 = \begin{bmatrix} -53.643 & -10.832 & -92.426 & -22.881 \end{bmatrix}$$
$$G = -0.002$$
$$H_0 = -1.4459$$
$$K = \begin{bmatrix} -0.3316 & 0.7418 \\ -18.6057 & 40.4547 \\ 0.1814 & -0.3322 \\ 8.7146 & -18.7541 \end{bmatrix}$$

C. PERFORMANCE VALIDATION SETUP BY NUMERICAL SIMULATIONS

Fig. 8 shows the setup of the WNCS in the numerical simulation. The feedback controller (Deep Learning BOX II, Gdep Advance Co., Japan) and the remaining apparatus were located in two distinct rooms on the campus. A notebook PC and a Raspberry Pi board were positioned beside the real pendulum manipulator. The notebook PC in the numerical experiments did not communicate with the in-house controller of the manipulator but did calculate the response of the system using the identified model as disclosed in the previous subsection. We connected the notebook PC and the Raspberry Pi board with a LAN cable in the numerical simulation. The response of the pendulum angle α was sent from the notebook PC to the Raspberry Pi via a UDP diagram. The notebook PC and the Raspberry Pi both constituted two sensor nodes. The notebook PC fed back the response of the manipulator joint $\theta(t)$, whereas the Raspberry Pi returned the angle of the pendulum $\alpha(t)$. Both sensor nodes used the wireless network connection, and the responses were transmitted to the controller via the WiFi access point in the room, as depicted in Fig. 8.

We calculated the plant responses to three different controllers in our numerical simulation:

(A) The conventional two DOF controller that does not take delays into account, as specified by (17).
(B) The dynamic state predictive controller proposed in our recent conference paper [23]. As the RNN-based RTT predictor implemented in this control law is trained to predict the behavior of RTT1, RTT2 is discarded in this feedback controller. The control law is specified by (18).

TABLE 1. Physical parameters of the system.

Pendulum	
Mass	$m_1 = 0.224$ kg
Distance from origin O_2 to the center of gravity	$l_1 = 0.594$ m
Moment of inertia around the center of gravity	$J_1 = 6.6 \times 10^{-3}$ kg $\cdot$m^2
Viscous friction coefficient	D = 0 kg $\cdot$ m^2 / s
Manipulator Parameters	
Distance between origins O_1 and O_2	$L_0 = 0.850$ m
Values determined by system identification	$a_1 = 35.18$ $a_2 = 361.7$ $b_1 = 361.7$

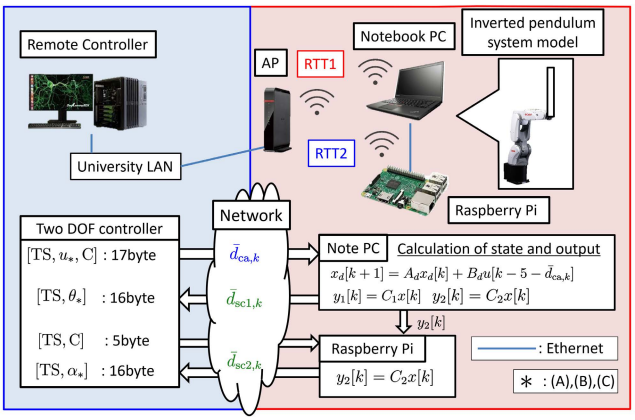


FIGURE 8. Setup of the numerical simulation for the WNCS.

(C) The proposed dynamic state predictive controller, as specified by (20).

Hereafter, the control inputs corresponding to controllers (A), (B) and (C) are denoted as u_A, u_B, and u_C, respectively. The feedback controller was used to calculate these three control inputs within a 5 ms sampling interval and transmit a 17-byte packet [TS, u_A, u_B, u_C, C] to the notebook PC, where TS represents the time stamp of the controller and C is a single byte character representing the status. The feedback controller also transmitted a five-byte packet [TS, C] to the Raspberry Pi.

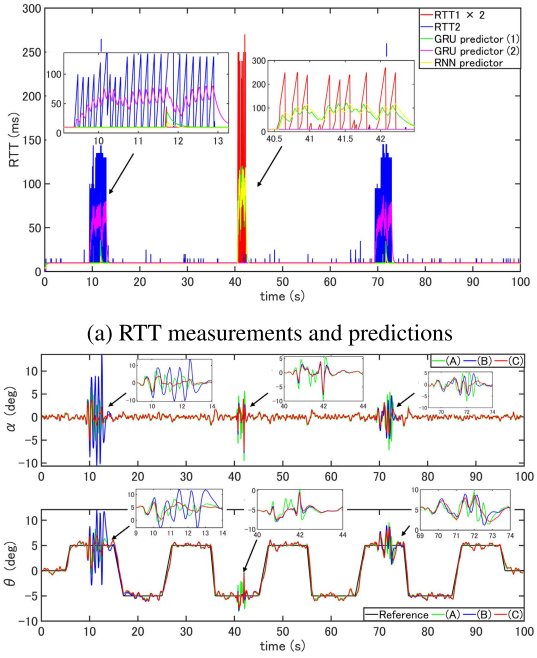
The notebook PC was used to calculate the responses of the plant modeled by (35) upon receiving a packet from the controller and transmit the pendulum angle to the Raspberry Pi. No transmission delay was assumed for this process because we used a wired network connection, and the network cable length was short. Once the Raspberry Pi received the pendulum angle, it added a random noise of $\pm 1°$ to simulate the measurement noise in the real environment. Then, the notebook PC returned a 16-byte packet [TS, θ_A, θ_B, θ_C] to the controller, and the Raspberry Pi computer also returned a 16-byte packet [TS, α_A, α_B, α_C]. Here, TS represents the timestamp included in the packet from the controller. Including TS in the return packets enabled the calculation of the RTTs of the two network communications that were necessary for RTT predictors to determine future RTTs.

Let RTT1 be the RTT between the controller and the notebook PC (Sensor1) and let RTT2 be the RTT between the controller and Raspberry Pi (Sensor2). We applied a Kalman filter defined by (27) to (30) with $r = 0.05$ and $q = 20$. In the implementation of the proposed controller (C), we calculated the RTT predictions for $H = 7$ steps ahead and determined $\hat{H}_{*,k}(* = 1, 2)$ using the Kalman filter. Because a constant delay of five sampling intervals exists in the TCP/IP communication between the notebook PC and in-house controller of the manipulator, we set $\hat{H}_k = 5$, even when we obtained a $\hat{H}_{*,k}$ value smaller than five.

D. SIMULATION RESULTS AND DISCUSSION

The RTT prediction results are reported in this section. Fig. 9a shows the measured and predicted RTTs when we used the 5 GHz frequency band connection. As disclosed in our investigation on wireless network delay in Section II-A, we observed that the RTT $\tau(t)$ in the experimental peer-to-peer network connection was not the summation of two E2E delays ($\tau(t) = \tau_1(t) + \tau_2(t)$), but they all took almost the same value ($\tau(t) \fallingdotseq \tau_1(t) \fallingdotseq \tau_2(t)$). We accordingly inferred that three network transmission delays $\bar{d}_{*,k}$, where * represents ca, sc1 or sc2, that appeared in (4) and (6) might also be nearly equal in our target WNCS depicted in Fig. 8 if the network was under a state of congestion.

Let us assume that the controller sends out a packet at time k during network congestion. If the packet takes D steps to be delivered to the notebook PC ($\bar{d}_{ca,k} = D$), the notebook PC returns a packet in response to the arrival of the delayed



(a) RTT measurements and predictions

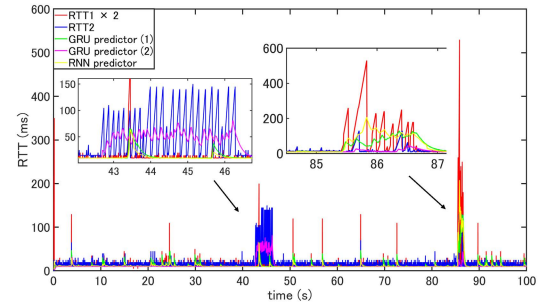
(b) Numerical responses of the pendulum and first joint of the manipulator

FIGURE 9. Numerical simulation results corresponding to the 5 GHz wireless connection.

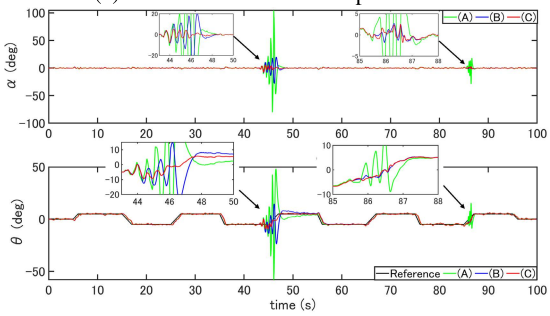
packet at time $k + D$. If the network is still under a state of congestion, as the returned packet includes the time stamp corresponding to k, not $k + D$, we presume that it will take an additional D steps for the packet to be delivered to the controller ($\bar{d}_{sc1,k+D} = D$). Accordingly, we observe RTT1 $= 2D$ in the worst case. This is the reason why we plotted the doubled RTT1 measurement in Fig. 9a. We used the doubled RTT1 measurements in the training of the GRU predictor for RTT1 and the calculation of the prediction. In contrast to the issues in the controller-Sensor1 network, we only expected a single trip E2E delay in the controller-Sensor2 network. We accordingly used the measured RTT2 values without the treatment as we did with RTT1 measurements, and the GRU network calculations were performed with the raw RTT2 measurements.

The plot labeled "RNN predictor" is the result of a one step ahead prediction of the RTT1 calculated using our previous RNN predictor that was developed in [23]. The RNN prediction was also smoothed using a Kalman filter. We observed that both the GRU and RNN RTT predictions varied almost in phase with their corresponding RTT measurements. However, the GRU predictions exhibited more in-phase behavior with the RTT measurements, whereas the RNN predictions suffered from a larger phase lag.

Next, we discuss the response of the proposed system model. The steady-state values of the reference r for the manipulator joint angle were set to $\pm 5°$. The actual reference trajectory was linearly interpolated in the transients (from $+5°$ to $-5°$, and vice versa) to avoid activating the software



(a) RTT measurements and predictions



(b) Numerical responses of the pendulum and first joint of the manipulator

FIGURE 10. Numerical simulation results corresponding to the 2.4 GHz wireless connection.

emergency brake of the in-house controller when we actually drive the manipulator later. The software emergency brake was used to cut out the motor motion immediately after detecting a large angular velocity.

Because the RTTs took the values of two to three sampling steps when the network was not congested, we did not observe significant differences between the three responses corresponding to the three control inputs. However, apparent differences were observed during network congestion. When the RTT1 exhibited large values (at approximately 40 s in Fig. 9a), both the manipulator and the pendulum suffered from large oscillations with the control input (A), which did not account for the transmission delay. The remaining two controllers succeeded in attenuating oscillations during network congestion. No significant differences were observed between the responses corresponding to the control inputs (B) and (C).

When the RTT2 exhibited large values, the proposed controller (C) only succeeded in achieving the two control objectives. It is worth noting that controller (B) failed to stabilize the plant when the discarded RTT2 took large values because of the out-of-phase predicted state of the pendulum angle $\alpha(t)$ and its derivative $\dot{\alpha}(t)$.

We then altered the wireless communication frequency band to 2.4 GHz and performed the same simulations. The perturbations of the measured RTTs in this frequency band were apparently larger than that of the 5 GHz band even when the network was not congested. However, as the RTTs remained within the stability margin guaranteed by the LMI-based gain when the network was not congested, the stability of the closed-loop control system was maintained, and we did not observe any significant difference in the three responses.

When the second sensor node network was congested (RTT2 was large), we found that the closed-loop system lost its stability with the controllers (A) and (B). The proposed controller (C) succeeded in maintaining closed-loop stability under network congestion of the two sensor nodes. As the discarded RTT2 in the controller implementation (B) corresponds to the measurement of the unstable part of the plant dynamics, the results imply that appropriate compensation of the measurement delay of the unstable state variables is very important for closed-loop stability.

The same inference can be applied to the extension of the proposed switching dynamic state predictive control to other WNCSs that have more than two network nodes. If there are $m(>2)$ network nodes included in the WNCS, we elaborate on formulating and training m GRU RTT predictors. The right-hand side of (20) will be extended to include m control law equations, each of which includes the predicted RTT $\hat{H}_{i,k}$ $(i = 1, 2, \cdots, m)$. A straightforward extension of (20) would be to choose the control input calculated with the worst case RTT of m network nodes. However, this approach might only work conditionally.

If the plant dynamics related to every sensor node are stable or if all the m RTT predictions are similar, an acceptable control performance can be expected. However, if a large difference exists in the m RTTs, or if more than two sensor nodes are related to the unstable plant dynamics, we cannot assert that the application of the extended version of the proposed dynamic state predictive control to other WNCSs would be successful. We currently believe that providing the in-phase prediction of unstable state variables may be the key to the successful application of the proposed control system to a WNCS with more than two sensor nodes.

E. ADDITIONAL CLOSED-LOOP STABILITY ANALYSIS OF THE DYNAMIC STATE PREDICTIVE CONTROL

From the results of the numerical simulations, it can be observed that the measured RTTs sometimes exceeded the stability tolerance that the LMI-based feedback controller gain F_0 guarantees; however, the proposed controller maintained the closed-loop stability for all setups tested. As the LMI conditions were derived by assuming the algebraic description of the control law (A), we analyzed the closed-loop stability when we used the dynamic state predictive controllers (B) and (C) under certain assumptions for simplicity.

We hereafter assumed that all states can be directly measured, and hence omitted the state observer in the rest of the development. We also assumed that $\bar{d}_{sc1,k} = \bar{d}_{sc2,k}$. Let H_k represent the predicted RTT at time step k. Then, the state predictive controller is defined as

$$u[k] = F_0 x[k + H_k] + H_0 r + v[k], \tag{36}$$

where the future state prediction $\hat{x}[k + H_k]$ should be expressed using the quantities available at time k. We achieved this by assuming that $\bar{d}_k > H_k$ holds, and we obtained

$$x[k+1] = A_d x[k] + B_d F_0 \left(A_d^{H_k} x[k - \bar{d}_k] + \sum_{j=k-H_k}^{k-1} A_d^{k-j-1} \times B_d u[j - \bar{d}_{\mathrm{ca},k}] + v[k - \bar{d}_k] \right). \quad (37)$$

Moreover, $x[k]$ can also be expressed as

$$x[k] = A_d^{H_k} x[k - H_k] + \sum_{j=k-H_k}^{k-1} A_d^{k-j-1} B_d u[j - \bar{d}_{\mathrm{ca},k}]. \quad (38)$$

By using this relation to eliminate the input-related terms in (37), we obtained

$$x[k+1] = (A_d + B_d F_0) x[k] + B_d F_0 A_d^{H_k} x[k - \bar{d}_k] - B_d F_0 A_d^{H_k} x[k - H_k] + B_d F_0 G \zeta[k - \bar{d}_k] \quad (39)$$

as the dynamics that govern the behavior of $x[k]$, where the adjoint quantity $\zeta[k]$ was governed by

$$\zeta[k+1] = \zeta[k] + F_1 B_d G \zeta[k - \bar{d}_k] - F_1 B_d F_0 A_d^{H_k} \times (x[k - H_k] - x[k - \bar{d}_k]). \quad (40)$$

The closed-loop dynamics of the state predictive control system were found to be

$$\xi'[k+1] = \Phi' \xi'[k], \quad (41)$$

where Φ' is the closed-loop matrix presented at the bottom of the page and $\xi'[k]$ is the corresponding state vector defined as

$$\xi'[k] = \left[x^T[k], x^T[k-1], \cdots, x^T[k - H_k], \cdots, x^T[k - \bar{d}_k], \zeta^T[k], \zeta^T[k-1], \cdots, \zeta^T[k - d_M] \right]^T.$$

As reported in Section V-B, we determined a solution of LMI with $d_M = 17$. If we set $\bar{d}_k = 30$ and assumed the corresponding estimate H_k to be 16 based on the observed worst case RTT2 value in Fig. 9a, the largest absolute value of the eigenvalues of Φ' was determined to be 0.9986, which is inside the unit circle of the complex plane. However, introducing $\bar{d}_k = 30$ to Φ in (15) results in the existence of an eigenvalue λ_0 outside the unit circle ($|\lambda_0| = 1.0077$), which justifies the instability of the closed-loop control system with control law of (A) observed in the numerical simulation.

Next, we considered the worst RTT1 value $\bar{d}_k = 50$, which amounts to a 250 ms delay in Fig. 9a. By introducing this value with $H_k = 23$ to Φ', we observed that there exists an eigenvalue λ_1 outside the unit circle ($|\lambda_1| = 1.0081$). However, when we introduced $\bar{d}_k = 50$ to Φ, the largest absolute value of an eigenvalue outside the unit circle was 1.017. This fact proves the improvement of the closed-loop stability when we apply a dynamic state predictive controller.

We believe that the reason why the closed-loop stability was maintained when we used the proposed dynamic state predictive controller (C), as shown in Fig. 9a, depends significantly on the time-varying behavior of the RTT in wireless network communication. As shown through the results of our RTT measurements, the extreme RTT values only lasted for a couple of samples. Instantaneous instability may have occurred in our simulation, as evidenced by the oscillatory behaviors shown in Figs. 9b and 10b. However, as the RTTs decreased considerably after their extremes, the stability of the control system was maintained.

F. HOW THE GRU CONFIGURATION AFFECTED THE RTT PREDICTION

We showed that the dynamic state predictive control scheme contributed to the closed-loop stability in our WNCS. As the prediction of RTT is key information for successful implementation of the proposed control system, we now discuss the configuration of the GRU RTT predictor.

Since our control objective includes the stabilization of the unstable inverted pendulum, high rate control calculations are expected for better control performance. At this point, we have to balance the trade-off between the computational cost of the predictor and the accuracy of RTT prediction. Adding more hidden layers to the GRU RTT predictor can be an attractive option for improved prediction accuracy at the expense of high rate control.

We accordingly decided to use one hidden layer GRU model for RTT prediction. As we can still alter the number

$$\Phi' = \begin{bmatrix} A_d + B_d F_0 & 0 & \cdots & 0 & -B_d - F_0 A_d^{H_k} & 0 & \cdots & 0 & B_d F_0 A_d^{H_k} & 0 & \cdots & 0 & B_d G \\ I & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & I & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & -F_1 B_d - F_0 A_d^{H_k} & 0 & \cdots & 0 & F_1 B_d F_0 A_d^{H_k} & I & \cdots & 0 & F_1 B_d G \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & I & 0 & 0 & \cdots & I & 0 \end{bmatrix}$$

TABLE 2. Prediction performance comparison of the three GRU networks.

Number of hidden layer neurons	RMSE of the seven steps ahead prediction of RTT2 in the 5 GHz wireless network	RMSE of the seven steps ahead prediction of RTT2 in the 5 GHz wireless network
50	63.73 ms	59.42 ms
85	14.37 ms	14.81 ms
120	11.65 ms	12.90 ms

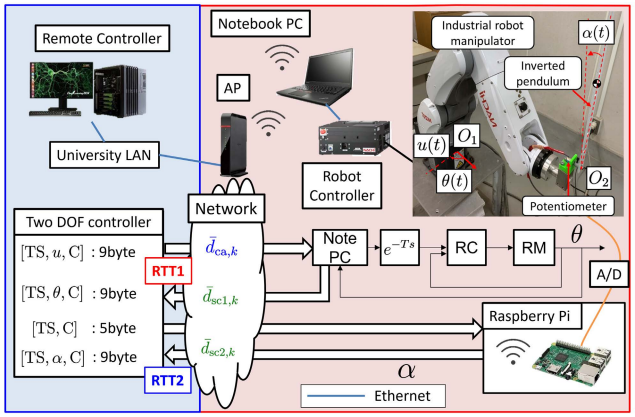
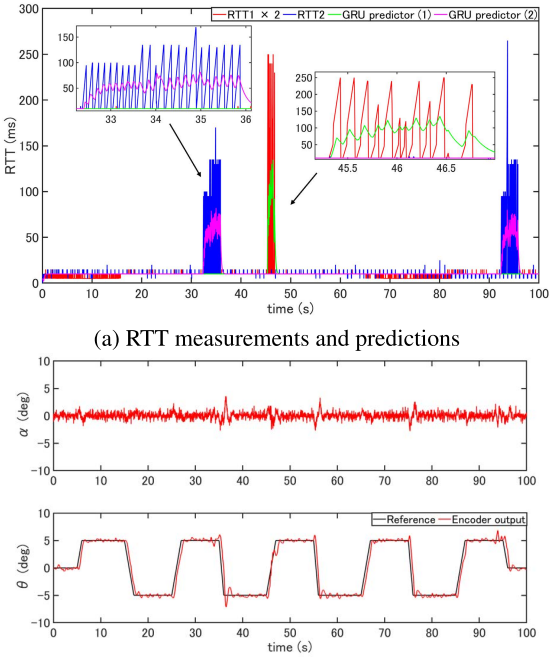


FIGURE 11. Setup of the manipulator-driven inverted pendulum stabilization experiment over a wireless network.



(a) RTT measurements and predictions

(b) Responses of the pendulum and the first joint axis of the manipulator

FIGURE 12. Experimental results corresponding to the 5 GHz wireless connection.

of hidden layer neurons for our single hidden layer GRU, we configured and trained two additional single hidden layer GRU RTT predictors that had 50 and 85 neurons, respectively. We trained these two models with the same training dataset as we used to train the GRU predictor with 120 neurons that was implemented in the proposed dynamic state predictive controller. The same set of hyperparameters as was used in the GRU predictor with 120 hidden layer neurons was used for the additional two GRU RTT predictors.

After the training was completed, we applied the measured RTT2 sequences as shown in Figs. 9a and 10a, to the predictors and calculated seven steps ahead prediction of RTT2 in the same manner as stated in Section IV-B. We calculated the root mean squared error for the seven steps ahead prediction of RTT2 as the prediction accuracy metric, and the results are summarized in Table 2.

This table shows that the GRU predictor with 50 hidden layer neurons failed to predict the seven steps ahead behavior of the RTT2, whereas the remaining two GRU predictors exhibited comparable accuracy metrics. A significant improvement in the accuracy was observed when we increased the neurons from 50 to 85. However, the improvement was less relevant when we increased the neurons from 85 to 120. We inferred that adding more hidden layer neurons might contribute little while increasing computational time, and we confirmed that using 120 hidden layer neurons was a reasonable choice.

VI. EXPERIMENTAL VALIDATION

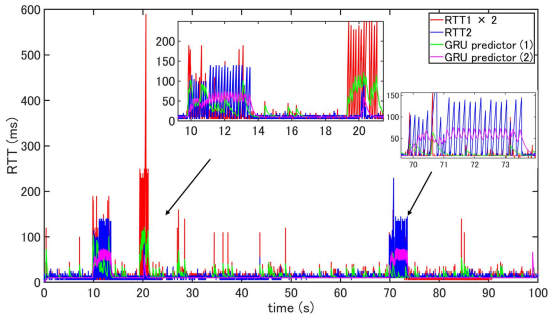
A. EXPERIMENTAL SETUP

In the next experiment, the proposed controller (C), which exhibited the best control performance in the numerical simulation, was implemented in a real inverted pendulum experiment with a robotic manipulator. Fig. 11 shows the experimental apparatus. We used the same network connection that was established in the numerical simulation. The numerical model was replaced with a real inverted pendulum and robotic manipulator. The packets communicated between the controller and the two sensor nodes were modified. The controller transmits a nine-byte packet of [TS, u, C] to the notebook PC, where u represents the control input calculated with the proposed control law. The controller also transmits a five-byte packet [TS, C] to the Raspberry Pi in every sampling interval $h = 5$ ms.

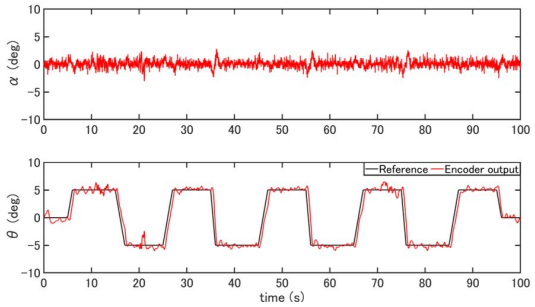
When the notebook PC received a packet from the controller, u was transmitted to the in-house controller of the manipulator using the TCP/IP cable connection. The notebook PC also received the control result from the in-house controller, retrieved the first-joint axis angle θ, and generated a nine-byte packet [TS, θ, C] to be transmitted back to the controller. The pendulum angle was measured using a potentiometer (CP-2UN, Midori Precisions Co., Japan), whose output was converted to an angle in the Raspberry Pi via a 12 bit analog-to-digital converter (MCP3208, Microchip Technology Inc., Arizona, USA) every 5 ms, and the Raspberry Pi returned a nine-byte packet [TS, α, C] to the controller.

B. RESULTS OF THE EXPERIMENT

The experimental results are summarized in two plots, as shown in Figs. 12 and 13 for the 5 GHz and 2.4 GHz



(a) RTT measurements and predictions



(b) Response of the pendulum and the first joint axis of the manipulator

FIGURE 13. Experimental results corresponding to the 2.4 GHz wireless connection.

wireless connections, respectively. When the network connection was stable, the responses were fine. When we considered the RTT1 or RTT2 under conditions of network congestion, we observed a marginally larger oscillation of the pendulum, which also affected the tracking control of the manipulator joint. However, the controller performed the two tasks as expected even under congestion, and we observed no steady-state error in the tracking task.

We concluded from the results of the experiments that the proposed switching-type dynamic state-predictive controller with the GRU-based RTT predictor performed as expected in the two DOF digital servo WNCS.

VII. CONCLUSION AND FUTURE WORK

In this paper, we described the synthesis of a two DOF digital servo controller design for a WNCS and proposed a switching-type control law with dynamic state prediction using the output of the GRU-based RTT predictor. We determined the feedback controller gains while considering the network transmission delay. The gains with the proposed control law guaranteed the closed-loop stability of the WNCS when the network was less congested and the tracking to the step changes of the reference. We proposed integrating the dynamic state predictive control into the aforementioned controller to improve the stability under network congestion, which is commonly observed in wireless network communications.

The results of the simulation of the pendulum manipulator WNCS and its experimental validation showed that the proposed switching-type controller exhibited good performance in a networked control environment where multiple network nodes exist. As discussed in Section V-D, the time-varying nature of the RTT in the NCS makes the theoretical treatment of the stabilization in controller design a challenging task. We would like to apply the current results to other WNCSs with more than two communication nodes and continue the development of controller synthesis to improve the closed-loop stability of wireless networked control problems.

ACKNOWLEDGMENT

The authors would like to thank American Journal Experts (www.aje.com) for their assistance with English language editing. They would also like to thank the associate editor and the anonymous reviewers for their helpful comments.

REFERENCES

[1] T. Kumagai, H. Yoshida, and K. Satoda, "Adaptive remote control of a mobile robot system with delay fluctuation," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2018, pp. 1–5.
[2] N. Becker, A. Rizk, and M. Fidler, "A measurement study on the application-level performance of LTE," in *Proc. IFIP Netw. Conf.*, Jun. 2014, pp. 1–9.
[3] O. J. M. Smith, "Closer control of loops with dead time," *Chem. Eng. Prog.*, vol. 53, no. 5, pp. 217–219, May 1957.
[4] Z. Palmor, "Stability properties of Smith dead-time compensator controllers," *Int. J. Control*, vol. 32, no. 6, pp. 937–949, May 1980.
[5] T. Furukuwa and E. Shimemura, "Predictive control for systems with time delay," *Int. J. Control*, vol. 37, no. 2, pp. 399–412, 1983.
[6] H.-J. Yoo and O.-K. Kwon, "Networked control systems design with time-varying delays," *IFAC Proc. Volumes*, vol. 38, no. 1, pp. 48–53, 2005.
[7] M. Garcia-Rivera and A. Barreiro, "Analysis of networked control systems with drops and variable delays," *Automatica*, vol. 43, no. 12, pp. 2054–2059, 2007.
[8] Y.-B. Zhao, G.-P. Liu, and D. Rees, "Stability and stabilisation of discrete-time networked control systems: A new time delay system approach," *IET Control Theory Appl.*, vol. 4, no. 9, pp. 1859–1866, 2010.
[9] Y.-B. Zhao, G.-P. Liu, and D. Rees, "Packet-based deadband control for internet-based networked control systems," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 5, pp. 1057–1067, Sep. 2010.
[10] A. Kuzu, S. Bogosyan, and M. Gokasan, "Predictive input delay compensation with grey predictor for networked control system," *Int. J. Comput. Commun. Control*, vol. 11, no. 1, p. 67, Nov. 2015.
[11] H.-C. Yi, C.-J. An, and J.-Y. Choi, "Compensation of time-varying delay in networked control system over Wi-Fi network," *Int. J. Comput. Commun. Control*, vol. 12, no. 3, pp. 415–428, Jun. 2017.
[12] H. Yoshida, T. Kumagai, and K. Satoda, "Dynamic state-predictive control for a remote control system with large delay fluctuation," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2018, pp. 1–6.
[13] G. P. Liu, Y. Xia, J. Chen, D. Rees, and W. Hu, "Networked predictive control of systems with random network delays in both forward and feedback channels," *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1282–1297, Jun. 2007.
[14] A. Gezer, "Large-scale round-trip delay time analysis of IPv4 hosts around the globe," *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 27, no. 3, pp. 1998–2009, May 2019.
[15] S. Belhaj and M. Tagina, "Modeling and prediction of the internet end-to-end delay using recurrent neural networks," *Proc. J. Netw.*, vol. 4, no. 3, pp. 528–535, 2009.
[16] Y.-B. Zhao, X.-M. Sun, J. Zhang, and P. Shi, "Networked control systems: The communication basics and control methodologies," *Math. Problems Eng.*, vol. 2015, pp. 1–9, Jun. 2015.
[17] T. Li, D. Leith, and D. Malone, "Buffer sizing for 802.11-based networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 1, pp. 156–169, Feb. 2011.
[18] A. G. Parlos, "Identification of the internet end-to-end delay dynamics using multi-step neuro-predictors," in *Proc. Int. Joint Conf. Neural Networks. (IJCNN)*, vol. 3, 2002, pp. 2460–2465.

[19] D. H. Hagos, P. E. Engelstad, A. Yazid, and C. Griwodz, "A deep learning approach to dynamic passive RTT prediction model for TCP," in *Proc. IEEE 38th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Oct. 2019, pp. 1–10.
[20] A. Dong, Z. Du, and Z. Yan, "Round trip time prediction using recurrent neural networks with minimal gated unit," *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 584–587, Apr. 2019.
[21] S. Yasuda and H. Yoshida, "Prediction of round trip delay for wireless networks by a two-state model," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.
[22] L. Lu, Q. Liang, Q. Zhu, and Y. Zhao, "Prediction of round trip delay for wireless networks by a two-state model," in *Proc. Chin. Automat. Congr. (CAC)*, 2020, pp. 2729–2734.
[23] T. Nonomura and F. Fujii, "Design of two DOF digital servo system with RNN-based dynamic state predictive control for network control systems with large delays," in *Proc. IEEE/SICE Int. Symp. System Integr. (SII)*, Jan. 2022, pp. 944–949.
[24] S. Belhaj and M. Tagina, "Modeling and prediction of the internet end-to-end delay using recurrent neural networks," *J. Netw.*, vol. 4, no. 6, pp. 528–535, Aug. 2009.
[25] T. Hagiwara, M. Ichiki, M. Kanaboshi, K. Fukumitsu, and M. Araki, "Digital two-degree-of-freedom LQI servo systems," *Trans. Inst. Syst., Control Inf. Eng.*, vol. 11, no. 2, pp. 51–60, 1998.
[26] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of Adam and beyond," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–23.
[27] S. Goto, M. Nakamura, J. Zuo, and N. Kyura, "Reduced order model configuration for respective operating speed of industrial mechatronic servo Systems," *J. Jpn. Soc. Precis. Eng.*, vol. 65, no. 6, pp. 815–819, 1999.
[28] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optim. Methods Softw.*, vol. 11, nos. 1–4, pp. 625–653, 1999, doi: 10.1080/10556789908805766.

TATSUKI NONOMURA received the B.E. degree from Yamaguchi University, Japan, in 2020, where he is currently pursuing the M.E. degree with the Department of Mechanical Engineering, Graduate School of Sciences and Technology for Innovation. His research interests include networked control systems, robotics, and machine learning.

FUMITAKE FUJII (Member, IEEE) received the B.E., M.E., and Ph.D. degrees from the Tokyo Institute of Technology, in 1992, 1994, and 2002, respectively. In 1994, he worked with the Japanese Patent Office and moved to Yamaguchi University, in 1995, where he started his work as a Research Associate. He is currently an Associate Professor with the Department of Mechanical Engineering, Yamaguchi University. His research interests include phenomenological modeling of dynamical systems, system integration in radiation therapy, and interaction in human–machine systems. He is a member of the Japan Society of Mechanical Engineers, Society for Instrument and Control Engineers, and Robotic Society of Japan.