International Journal of Control, Automation, and Systems (2014) 12(1):102-110
DOI 10.1007/s12555-013-9156-2

ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Experimental Analysis and Implementation of a Multiscale Wireless/Wired Networked Control System

Jiawei Dong and Won-jong Kim*

Abstract: The aim of this paper is to discuss the flexibility and the performance of a multiscale wireless/wired networked control system (NCS). This NCS consists of three different types of dynamic systems (fast, medium, and slow clients) with distinct time scales. The experimental results verified the capability of the NCS to combine both the wired and wireless networks and the control capability of the NCS with various sampling periods. Compared to the original wired NCS, the average steady-state error of the fast client increased by 20% to 30% under the same conditions with the wireless NCS within the bandwidth-utilization (BU) threshold. From the analysis, the sampling period together with the BU and the number of clients will determine the time-delay and packet-loss levels and affect the stability and performance of the NCS in a complex correlated manner.

Keywords: Bandwidth utilization, multiscale networked control system, real-time system, wireless and wired networks.

1. INTRODUCTION

In the last several decades, NCS has been employed in various applications such as mobile sensor networks, remote surgery, haptics collaboration, and unmanned aerial vehicles [1-4]. Although the NCS has the merits of reduced weight and space requirement, ease of system diagnosis and maintenance, and increased system agility, the spatial orientation of an NCS application could be limited by its physical length of the communication medium. However, this limited NCS framework could be expanded thanks to the rapid development of the wireless network.

Wireless networks for control applications have currently been envisioned with the existing network protocols such as Bluetooth, ZigBee, and Wireless Local Area Network (WLAN), etc [5]. In [6], Boughanmi *et al.* discussed that the quality of control (QoC) could be affected by inherent network factors, such as loads of traffic, delays, and jitters, in the case of a multi-hop control loop using WLAN. A bidirectional teleoperation mobile robot with force-feedback control was proposed in [7]. The mobile robot was controlled over the Bluetooth, and the visual information was fed back over the wireless network. A nonholonomic wheeled mobile manipulator was proposed in [8], formed by mounting the manipulator arms on disc-wheeled mobile bases and controlled with a dynamic level redundancy resolution strategy using a wireless Ethernet bridge. In [9], an intelligent cooperative control and a path-following algorithm were discussed based on a fuzzy model. The experimental results of three mobile robots traveling on different paths were presented to show the accuracy of obtaining control and cooperation by using the fuzzy algorithm with a ZigBee network. In [10], Wang *et al.* discussed the performance of an experimental vehicular wireless system that was applied for intelligent transportation system. Zhu *et al.* investigated the field performance of mobile sensing nodes developed for system identification and condition monitoring of civil structures in [11]. Each node consisted of a wall-climbing robot capable of navigating on steel structures, measuring structural vibrations, processing measurement data, and wirelessly communicating information.

When a wireless network introduces significant merits to an NCS such as flexibility, the challenge is also inevitably added to the complexity of the traditional wired NCS. Two major wireless standards, IEEE 802.11 and Bluetooth, were studied to highlight the performance aspects that were relevant to the control and automation networks in [12]. The performance of wireless Control Area Networks (CAN) and the latency were evaluated in [13]. Tabbara *et al.* proposed an Lp stability and scheduling protocol of wireless and wired NCSs in [14]. This scheduling protocol was implemented in several applications. Although the stability and the scheduling protocol were mainly developed on a wired NCS, they could also be applied to a wireless NCS. Ungan designed a PID controller from a wired NCS and then applied it to a wireless NCS in [15]. Zhang *et al.* discussed the predicative performance evaluation of the NCS in [16]. An optimal controller that minimizes the performance index was proposed. In [17], the authors gave a thorough study focusing on network imperfections and perform-

Manuscript received April 1, 2013; revised July 30, 2013; accepted August 22, 2013. Recommended by Associate Editor Hamid Reza Karimi under the direction of Editor Myotaeg Lim.

This work was supported in part by the TAMU Program to Enhance Scholarly and Creative Activities under Grant No. 2010-SAC-8779.

Jiawei Dong and Won-jong Kim are with the Department of Mechanical Engineering, Texas A&M University, College Station, TX, U.S.A. (e-mails: jwdong@neo.tamu.edu, wjkim@tamu.edu).
* Corresponding author.

ance of the NCS including time delays, packet losses and disorders, time-varying sampling periods, etc. Guan *et al*. studied optimal tracking performance issues for multi-input-multi-output linear time-invariant systems under networked control with limited bandwidth in [18].

In the aforementioned references, the wired and wireless NCSs were not combined together. Although the NCS is an active research field in academia and industry, the wired and wireless NCSs are investigated separately because of the structure and complexity concerns. Tabbara *et al*. and Ungan studied both the wired and wireless frameworks within a single NCS, but the analysis was at the simulation level [14,15]. The authors applied either one single controlled system or several identical controlled systems in the NCS. In practice, however, an NCS may include quite heterogeneous systems that require various control efforts. A physical environment requires that an NCS can handle not only a wired network but also a wireless network simultaneously with the ability to control various clients with different sampling-period requirements. If an NCS can only control one type of systems at a time, the cost will increase. The expandability will also suffer when the NCS is applied to large-scale industrial or distributed applications with many subsystems. To our best knowledge, few papers have been published to date to discuss either the performance of multiscale NCS or the NCS combined with wired and wireless networks. The objective of this research is to discuss the feasibility and performance of the multiscale wireless/wired NCS in a practical setting. For the research presented in this paper, a wireless robotic wheelchair is introduced to a wired NCS that consists of a ball magnetic-levitation (maglev) system and a DC motor speed-control system.

This paper is organized as follows. The real-time operating environment of the entire NCS is presented in Section 2. The experimental results of the NCS with and without the wireless client are provided in Section 3, followed by the conclusions in Section 4.

2. REAL-TIME OPERATING ENVIRONMENT

The NCS in this paper includes a ball maglev system, a DC motor speed-control system, and a wireless autonomous robotic wheelchair as the clients. The entire architecture of this NCS is shown in Fig. 1. All the three clients share the same server by competing for the computation and network resources to maintain their stability and performance. The wired and wireless TAMULink is the data-exchange medium at Texas A&M University. The wired TAMULink is an Ethernet Local Area Network (LAN) with the IEEE 802.3 standard, and the wireless TAMULink is a WLAN with the IEEE 802.11 standard.

As shown in Fig 1, Clients 1 and 2 representing the wired clients in the NCS are connected to Server via a LAN. Client 3 includes a robotic wheelchair and a laptop that sends and receives data packets over the wireless network. Since the laptop runs a Windows operating system (OS) that cannot communicate with Linux OS directly, a gateway including a computer operated as an intermediary is set in the NCS.

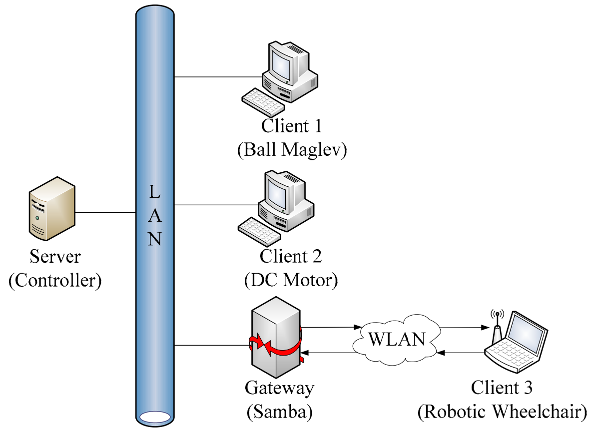


Fig. 1. The NCS architecture with three clients.

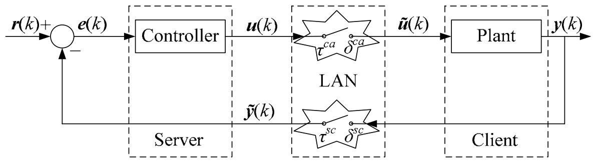


Fig. 2. NCS control block-diagram.

A typical closed-loop structure of an NCS is shown in Fig. 2. τ^{ca} and τ^{sc} represent the random time delays, and δ^{ca} and δ^{sc}, the packet losses in the controller-to-actuator and the sensor-to-controller links, respectively. In Fig. 2, $\mathbf{u}(k)$ and $\mathbf{y}(k)$ are the control inputs and plant outputs, respectively. $\tilde{\mathbf{u}}(k)$ and $\tilde{\mathbf{y}}(k)$ are the delayed control inputs and plant outputs. $\mathbf{e}(k)=\mathbf{r}(k)-\tilde{\mathbf{y}}(k)$ is the error, and $\mathbf{r}(k)$ is the reference command.

2.1. Hardware setup

Client 1 is the ball maglev system shown in Fig. 3, and the details can be referred to [19]. In order to levitate the steel ball at a predetermined steady-state equilibrium position with an electromagnet, the ball maglev system consists of a personal computer (PC), a position sensor, a pulse-width modulator (PWM) power amplifier and power supplies to drive the light bulb and the electromagnetic actuator. The optical position sensing unit consists of an incandescent light source, a CdS photocell, and a 15-V DC power supply.

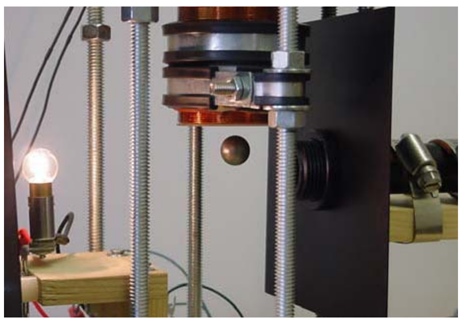


Fig. 3. Ball maglev system.

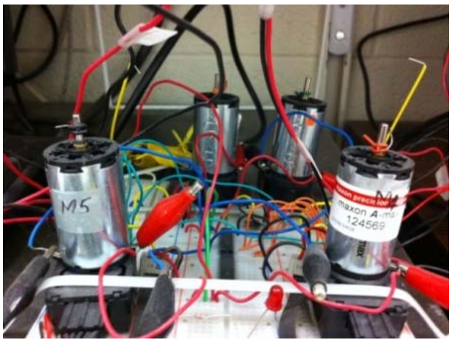


Fig. 4. DC motor speed-control system.

Fig. 5. Wireless autonomous robotic wheelchair.

Client 2, the DC motor speed-control system, is shown in Fig. 4 [19]. The speed of the DC motor is directly proportional to the supplied voltage, which is fed to the PWM amplifier. This drives the motor at a speed depending on the commanded voltage. The shaft angular displacement per unit time of the motor is sampled using the encoder. A PCI-6221 data-acquisition (DAQ) card by National Instruments (NI) enables the test bed to send out sensor-output data packets and receive control-input data packets through the LAN.

Client 3, the wireless autonomous robotic wheelchair, is shown in Fig. 5 [19]. Two independent 12-V DC motors are the actuators to drive the front wheels. The speed of the motors is controlled by the output voltage of the PWM amplifiers on the board of two Diverse Electronic's modular MC-7 motor controllers. An NI USB-6501 DAQ card performs all data-acquisition and control functions.

2.2. Software setup

To ensure the real-time operation of Server, Linux with Real-Time Application Interface (RTAI) is found as a competitive OS environment [20]. Linux Redhat 7.3 with RTAI 3.4 is chosen to be the OS running on Server. The Client 3 program was developed by Visual C++ 2008 with the Microsoft Windows XP OS, while the Clients 1 and 2 programs were developed by C on Ubuntu 6.10 with RTAI 3.4. The Client 3 program is built on Windows XP whereas Server program is built on Linux. Since Windows and Linux cannot communicate directly, Samba [21] was chosen to be a middle-ware. Samba is a standard Windows interoperability suite of programs for both Linux and UNIX.

Server	802.3 Header (14 bytes)	IP Header (20 Bytes)	UDP Header (8 Bytes)	Timestamp (8 Bytes)	Identifier (8 Bytes)	Control Data (12 Bytes)	SP (8 Bytes)

Client	802.1x Header (30 bytes)	IP Header (20 Bytes)	UDP Header (8 Bytes)	Timestamp (8 Bytes)	Identifier (8 Bytes)	Sensor Data (28 Bytes)	BU (8 Bytes)	Type (1 Bytes)

Gateway to Server	802.3 Header (14 bytes)	IP Header (20 Bytes)	UDP Header (8 Bytes)	Timestamp (8 Bytes)	Identifier (8 Bytes)	Sensor Data (28 Bytes)	BU (8 Bytes)	Type (1 Bytes)

Gateway to Client	802.1x Header (14 bytes)	IP Header (20 Bytes)	UDP Header (8 Bytes)	Timestamp (8 Bytes)	Identifier (8 Bytes)	Control Data (12 Bytes)	SP (8 Bytes)

Fig. 6. Data frame structures.

User Datagram Protocol (UDP) provides a datagram service that emphasizes reduced latency over reliability. It does not guarantee the datagram to be delivered to the destination host, and the datagram can also be delivered in an incorrect order. Although UDP is unreliable, it has fewer overheads than Transmission Control Protocol (TCP). This makes UDP much faster compared to TCP; it introduces less time delay than TCP does. Depending on the specifications and system requirements of various NCSs, UDP can be a possible choice as a suitable protocol. For some NCSs, UDP is a preferred protocol for better performance [22,23]. Due to the real-time characteristics of our NCS, UDP is chosen to be the protocol for the experiments.

2.3. Data-packet structures

The data-packet structures of Server, Clients, and Gateway are shown in Fig. 6. The 802.3 header, 802.11 header, IP header, and UDP header are the standard protocol headers. Control data and sensor data segments are the data segments generated by Server and Clients, respectively. Timestamp is set up by Clients to track the total time delays. Identifier is to identify Clients. The BU segment contains the current BU information of Clients. Type segment is used to identify whether Clients have a fixed sampling frequency or a variant one. The SP segment is the new sampling period to each client if applicable.

2.4. Network-induced time delays

As described in Fig. 1, the LAN and the WLAN were chosen to be the data-exchange medium. However, the use of the LAN and the WLAN poses several technical challenges including network time delays, packet losses, etc. To better understand the time-delay issue, consider one control iteration on the WLAN as shown in Fig. 7. The LAN follows the same timing diagram as the WLAN without the gateway in Fig. 7. Table 1 gives the nomenclature of the timing components.

Fig. 7 also illustrates the details of the communication in the NCS. In the beginning of the NCS experiments, Server waits for the data packets from either Client or Gateway after the socket setup. Client collects sensor feedback from the controlled plant and capsulates the data segment with necessary header segments into a single packet that is ready for transmission as shown in Fig. 6. The data packet will be transmitted if the network is idle or be held in a queue if the network is busy. If no packet loss takes place, the data packet will be transmitted through the network to its destination node

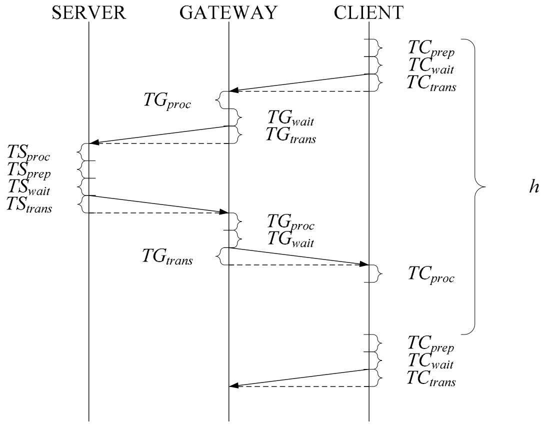


Fig. 7. Time delay components of the network in a periodic control loop.

Table 1. Nomenclature of the timing components.

Symbol	Description
T_{*prep}	Time taken by the client, the server, or the gateway to prepare the request message.
T_{*wait}	Time taken by the client, the server, or the gateway to wait for the network access.
T_{*trans}	Transmission time from the client, the server, or the gateway to its destination.
T_{*proc}	Time taken by the client or the server to process the message.
h	One sampling period of the client.

$*$ can be C (client), S (server), and G (gateway).

with a certain amount of propagation delay. The destination node will decode the data packet and implement the calculations. This process achieves the data transmission and calculation from Client to Gateway. The other transmissions in Fig. 7 follow the same steps. For Clients 1 and 2, Gateway is unnecessary.

As shown in Fig. 7, the control iteration has a sampling period h. The entire control is required to be completed in one sampling period, otherwise the data packets will be assumed to be lost or discarded. The total execution time τ of one single control iteration is given by

$$\begin{aligned}\tau = {} & TC_{prep} + TC_{wait} + TC_{trans} + TG_{proc} + TG_{wait} \\ & + TG_{trans} + TS_{proc} + TS_{prep} + TS_{wait} + TS_{trans} \\ & + TG_{proc} + TG_{wait} + TG_{trans} + TC_{proc}.\end{aligned} \tag{1}$$

In Table 1, the preparation times, the waiting times, and the transmission times are introduced by the network. The processing times are the time intervals for Clients, Server, or Gateway to process all data packets. From this point of view, the preparation times, the waiting times, and the transmission times can be classified as the network-induced time delays. And the processing times can be classified as the operation time.

Equation (2) gives a simple way to measure the execution time of each control loop.

$$\tau = T_{CP} + T_{PS} + T_P, \tag{2}$$

where T_{CP} is the round-trip time (RTT) from Client to the Server via Gateway, which includes TC_{prep}, TC_{wait}, TC_{trans}, TG_{wait}, and TG_{trans}. T_{PS} is the RTT from Server to Client via Gateway, which includes TS_{prep}, TS_{wait}, TS_{trans}, TG_{wait}, and TG_{trans}. T_P is the processing time that includes TS_{proc}, TC_{proc}, and TG_{proc}. If no gateway exists in the communication channels, the TG_* will be assumed to be 0, where * can be *wait*, *trans*, and *proc* as defined in Table 1. Note that τ can be random with respect to the control iterations due to the stochastic nature of the LAN and the WLAN, but will be assumed to be constant for each iteration in the experiments.

Table 2. RTT of the autonomous robotic wheelchair.

No.	Bytes	RTT (ms)		
		Min	Max	Average
S-G	56	0.765	1.600	0.908
G-S	56	0.756	2.262	0.938
C-G	56	1	52	7.1
G-C	56	1.371	26.086	7.533

Table 3. Processing time (ms) of the server, gateway and client.

No.	Server	Gateway	Client	Total
1	0.107	1.312	9	10.419
2	0.110	1.125	10	11.253
3	0.110	1.115	10	11.225
4	0.101	1.112	9	10.213
5	0.112	1.114	9	10.226
Average	0.108	1.156	9.400	10.660

Following (2), Tables 2 and 3 show the details of all the RTT and the processing time of the wireless wheelchair. The RTT tests were conducted with the PING command with 56 bytes for 5 rounds, each round with 1000 times. Table 3 shows the average results of the 5 rounds for each individual test. In Tables 2 and 3, the time resolutions are 1 microsecond (μs) on Linux and 1 millisecond (ms) on Windows. S-G represents Server-to-Gateway, and the other notations follow the same pattern.

Therefore, the total execution time of the wheelchair for each sampling period is

$$\begin{aligned}\tau &= T_{CP} + T_{PS} + T_P \\ &= \frac{0.908 + 0.938 + 7.1 + 7.533}{2} + 10.660 \\ &= 18.90 \text{ ms.}\end{aligned} \tag{3}$$

Similarly, the execution times of Clients 1 and 2 are 1.350 ms and 1.360 ms, respectively.

From [24], the relation between the sampling period and the BU can be indicated by the following equation,

$$b_i^k = \frac{\tau_i^k}{h_i^k}, \tag{4}$$

where b_i^k is the BU, h_i^k is the sampling period, and τ_i^k is the execution time. The subscript i indicates the index of Client in the NCS, and the superscript k indicates the control iteration.

From (4), a large sampling period implies a small BU, which gives more bandwidth available for other clients or other users in the network that are not parts of the

NCS. A small sampling period implies a large BU. Certain data packets would be dropped by the network protocol, or a larger time delay would be introduced to the NCS if the BU approaches a certain threshold, the saturation level of the network. If the BU exceeds the threshold, the performance or even the stability of the clients will be degraded because of the non-schedulability of all the real-time tasks. The BU threshold depends on the scheduling algorithm implemented by the scheduler, the number of clients in the NCS, the network conditions, etc.

3. EXPERIMENTAL RESULTS

As shown in Fig. 1, three clients are involved in the NCS experiments. Client 1, the ball maglev system, aims to levitate a steel ball at a 4-mm equilibrium position, which is measured from the bottom of the electromagnet to the top of the steel ball. Client 2, the DC motor speed-control system, aims to maintain the speed of the motor at 10 revolutions per second (rps). Client 3, the wireless autonomous robotic wheelchair, aims to explore an unknown environment with its real-time path planning capability. From the dynamics of each client, these three clients require various levels of sampling rates to maintain their stability. Client 1 requires a fast sampling rate; Client 3, a slow sampling rate; while Client 2 is in between. Refer to Table 4 in Section 3.2 for the actual sampling periods used in the experiments. This structure defines the NCS a multiscale system due to the various requirements from the combination of fast, medium, and slow systems, which brings more challenges to the NCS.

The number of clients in the NCS is not strictly limited to three. However, the complexity of the NCS will be increased by adding more clients. The maximum number of clients could be determined by the available network bandwidth and the minimum BU requirement of each client. For instance, if each client requires 30% network bandwidth, the NCS could incorporate three clients with 10% idle network bandwidth.

The performance of an NCS depends on the time-delay and packet-loss levels and bandwidth allocation and scheduling. The number of clients may not have direct impacts on the NCS, but it is indeed an important parameter of an NCS in the sense that more clients would compete for the network bandwidth and decrease the schedulability. In this case, a more robust controller may be necessary to the NCS, but the basic structure of the NCS will not change. Note that Client 1 is an open-loop unstable system that requires a fast fixed sampling rate to maintain its stability. Client 2 is an open-loop stable system that can have variant sampling rates. Therefore, the NCS in this paper includes Clients that are either wired or wireless systems, and open-loop unstable or stable systems with fast, medium, and slow dynamics. These Clients can have fixed or variant sampling rates. All these configurations bring more complexity to the NCS. In this paper, our objective is to verify the feasibility and the performance of the multiscale NCS with wired and wireless networks.

3.1. Control system design

The ball maglev system is an open-loop unstable system and requires a fast sampling rate to guarantee the stability and the performance. For this reason, a 3-ms sampling period is assigned to the system with the following controller that derived from the autoregressive (AR) model in [24]

$$\begin{aligned} u(k) &= 0.782u(k-1) + 0.13u(k-2) - 41500.0e(k) \\ &\quad + 48779.1e(k-1) - 31913.5e(k-2), \end{aligned} \tag{5}$$

where $u(k)$ is the control input and $e(k)$ is the error. The discrete-time controller of the DC motor is as follows [24]

$$\begin{aligned} u(k) &= u(k-1) - (1.5 - 2.5h_2)e(k) \\ &\quad + (1.5 + 2.5h_2)e(k-1), \end{aligned} \tag{6}$$

where h_2 is the sampling period given in Table 4. This paper mainly focuses on the verification of the capability of the NCS with both wired and wireless networks, and the details of the ball maglev system and the DC motor are omitted without loss of generality.

The flow chart of the control algorithm on Server is illustrated in Fig. 8. The three clients send the sensor data to and request the control inputs from Server. Server responds all the requests by their scheduled sequences, priorities, and identification numbers. If the total BU (TBU) of the NCS is less than 100% or the BU threshold,

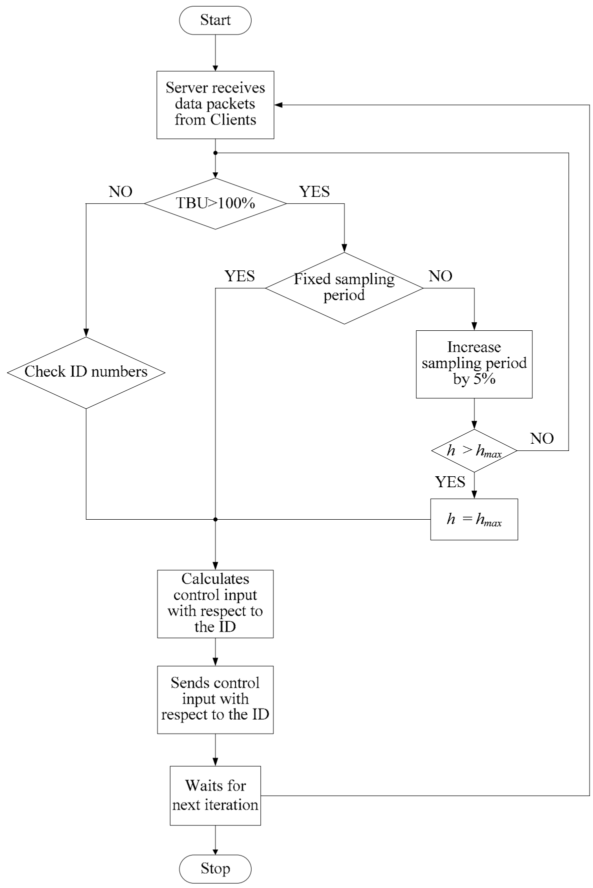


Fig. 8. Flow chart of the multiscale NCS control architecture.

Server calculates the control inputs and sends the control data packets to each client directly or via Gateway if applicable. If the TBU is greater than 100% or the BU threshold, Server checks the client's sampling period type and will calculate the control inputs directly for the clients that have fixed sampling periods. If the clients can have variant sampling period, Server will increase their current sampling periods by 5%, and check the TBU again until the TBU is no longer greater than 100% or the BU threshold. To maintain the stability of each client, the maximum sampling period h_{max} will be set as the boundary of the dynamic sampling period algorithm in the NCS.

3.2. Experiments

As discussed in Section 2, the BU threshold of the NCS depends on the implemented scheduling algorithm. In this paper, the Earliest Deadline First (EDF) is adopted for the experiments, which gives the NCS 100% BU threshold [25]. Table 4 presents four different combinations of BU and their corresponding sampling periods (h_i) and BU (b_i) of each client in the NCS. In Table 4, Case 4 exceeds the BU threshold. All the other three cases are within the BU threshold.

Figs. 9-12 illustrate the system performance of each client for the four cases of experiments in Table 4, respectively. Each figure shows the performances of the ball maglev system, the DC motor, and the wireless wheelchair as parts (a), (b), and (c), respectively. Although the TBU of Case 1 did not exceed the threshold, the performance of each client was degraded compared to Cases 2 and 3 as shown in Figs. 9, 10 and 11, respectively. It is because the sampling periods of Clients 2 and 3 in Case 1 were smaller than the one in Cases 2 and 3. Therefore, more data packets were exchanged in the network. It would introduce more time delays or even packet losses to the entire NCS such that the system performance could be degraded compared to the larger sampling-period cases. Packet loss may take place because the buffers in the network have limited capacities, and they will drop the data packets that cannot be scheduled.

From Fig. 12, the TBU was greater than 100% when Client 3 joined the experiment around 2 s. Based on the algorithm in Fig. 8, the sampling periods of Clients 2 and 3 increased by 5% each time until the TBU was no longer greater than 100%. Note that in Fig. 12 there was a performance degradation of Client 2 around 2 s when Client 3 joined the NCS. The sampling periods of Clients 2 and 3 were eventually reset as 3.83 ms and 102.10 ms, respectively. The TBU was reduced to 97.53% for Case 4.

Table 4. Four cases of experiments with the corresponding sampling periods h_i (ms) and BUs b_i (%).

Case	Client 1		Client 2		Client 3		TBU
	h_1	b_1	h_2	b_2	h_3	b_3	
1	3	43.5	5	27.0	100	18.95	84.95
2	3	43.5	10	13.5	150	12.63	69.63
3	3	43.5	15	9.0	300	6.3	58.8
4	3	43.5	3	45.0	80	23.6	112.1

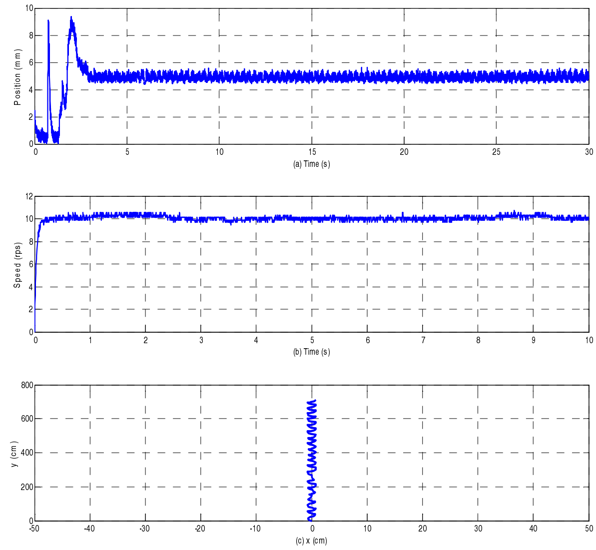


Fig. 9. Client motion trajectories from Case 1.

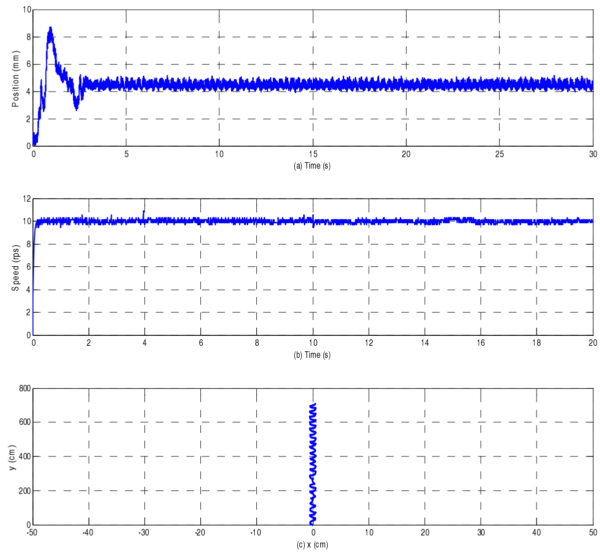


Fig. 10. Client motion trajectories from Case 2.

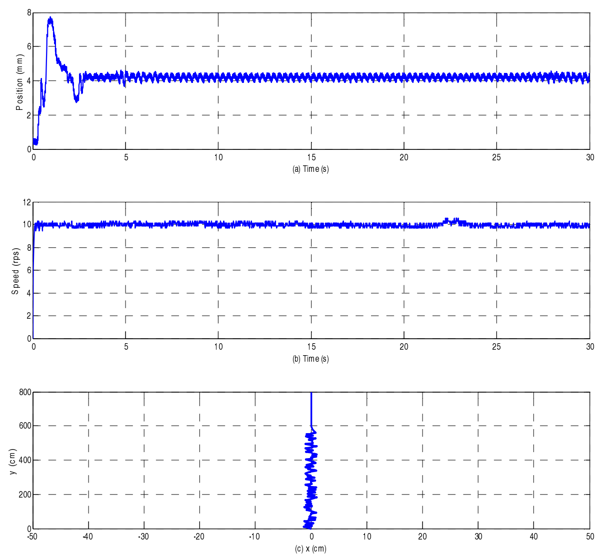


Fig. 11. Client motion trajectories from Case 3.

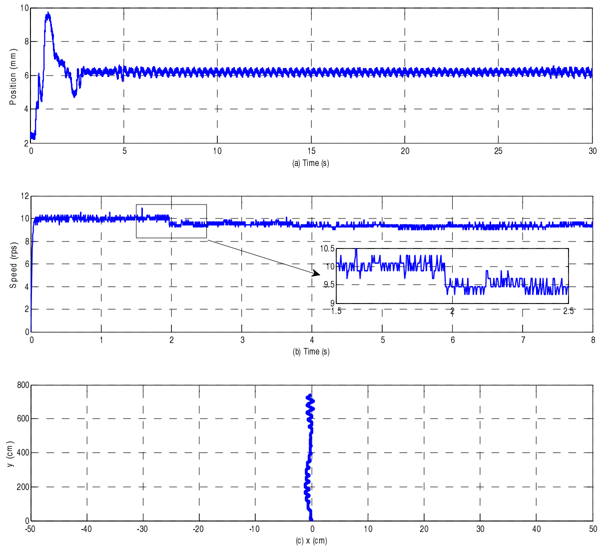


Fig. 12. Client motion trajectories from Case 4.

Table 5. System performance comparisons of NCS with wireless client.

Case		1	2	3	4
Client 1 (mm)	avg	0.7205	0.4851	0.2015	0.7653
	stdev	0.4053	0.2860	0.2261	0.4158
Client 2 (rps)	avg	0.0590	–0.0272	–0.0371	–0.0868
	stdev	0.2010	0.1673	0.1745	0.1869
Client 3 (cm)	avg	–0.0625	–0.0311	–0.1698	–0.0618
	stdev	0.5529	0.2861	0.7320	0.4920

Table 5 shows the performance comparison of these four cases of the NCS. The numbers in Table 5 are the averages and the standard deviations of each client. For Client 1, when the total BU increased by 10%, the average steady-state error increased by about 200%. For the other two clients, the medium client and the slow client, both the BU and the sampling periods affected the stability and the performance of the systems, but not as crucially as they were on the fast client, Client 1. If the TBU exceeds the BU threshold, the algorithm would bring the TBU of the NCS less than the threshold based on the type of each client's sampling frequency.

To determine the impacts of the wireless client to the NCS, the experiments, Cases 1 to 4, were performed with only wired clients under the same BU and network conditions as in Table 4. Another DC motor speed-control system was introduced to be Client 4, replacing the wireless robotic wheelchair, Client 3. In this paper, our intention is to analyze the impacts on the NCS from the wireless client, therefore, the same BUs were maintained as in Table 4, and Client 4's sampling periods were set as 7 ms, 10.7 ms, 21.4 ms, and 5.72 ms for Cases 1 to 4, respectively. Note that Client 4 had exactly the same system configuration as that of Client 2. The execution time of Client 4 is about 1.345 ms, which could be assumed as the same as Client 2. We would expect that Clients 2 and 4 have similar time responses if they are given the same sampling periods. Clients 2 and 4 have exactly the same system configuration and execution time, so Server will treat them equally. Although

Table 6. System performance comparisons of NCS without wireless client.

Case		1	2	3	4
Client 1 (mm)	avg	0.5635	0.3858	–0.1286	0.5325
	stdev	0.1402	0.1308	0.0916	0.1397
Client 2 (rps)	avg	–0.4556	–0.4370	–0.2865	–0.4275
	stdev	0.2269	0.2660	0.1969	0.2374
Client 4 (rps)	avg	0.2178	0.2525	0.0617	0.3104
	stdev	0.2569	0.2686	0.1913	0.2143

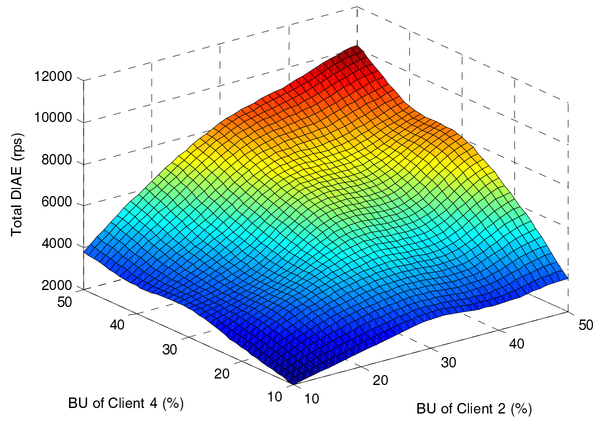


Fig. 13. Total DIAE vs. BUs of Clients 2 and 4.

Clients 2 and 4 are identical plants, the data packets from each client may not arrive at Server at exactly the same moment. In practice, however, the data packets will be queued up in Server's buffer waiting for unpacking and calculation. Based on this mechanism, Server will tell the sequence of the data packets from two or more identical clients. If this buffer sequence is not the expected one, different priorities can be assigned to the identical clients to rearrange their to-be-executed sequence.

Table 6 shows the performance comparison of these four cases of the NCS without wireless clients. The values in Table 6 follow the notations in Table 5. Comparing Tables 5 and 6, one can see that the BU has more crucial impacts on the system stability and performance as the dynamics of the system gets more complex. In the NCS without wireless clients (refer to the data given in Table 6), for the fast client, Client 1, with the same BU as in the NCS with a wireless client (Table 5), the average steady-state error decreased by about 20% to 30% within the BU threshold. The wireless network indeed introduced more complexity to the NCS. For the medium client, Clients 2, the average steady-state error increased because of the similar levels of the sampling periods as Client 4 although the BU of the NCS was exactly the same. Without a wireless client, Clients 2 and 4 competed for the resources more fiercely than the case with wireless client.

To show details of the discrete integral and absolute error (DIAE) versus BU of Clients 2 and 4, separate experiments were conducted. Because of the uncertainties and the time delays on the network, five sets of experiments were conducted with 20,000 times for each given BU. Each experiment varied the BU of Clients 2 and 4 from 10% to 50%. The average of the total DIAE of

Clients 2 and 4 is shown in Fig. 13. From Fig. 13, the DIAEs of Clients 2 and 4 are distributed evenly, which verifies the earlier analysis.

4. CONCLUSIONS

The multiscale NCS presented in this paper contains three clients, defined as the fast, medium, and slow client, respectively. With the wireless capability brought by WLAN, the NCS expands its flexibility at the cost of complexity to its frame structure. The wireless client has to compete for the resources with other wired on-board clients to guarantee its stability and performance. The experimental results verified the robustness of the NCS containing both the wired and wireless networks, and also the capability of the control of the NCS with various clients that require different sampling rates. Compared to the wired NCS, the average steady-state error of the fast client increased by about 20% to 30% under the same experimental conditions with a wireless NCS. Although all the data packets could be scheduled, some random data-packet losses would not destabilize the fast client. Meanwhile, the performance of the wired clients, such as the two DC motors, Clients 2 and 4, could be degraded drastically when two or more clients were employed at the same levels of sampling periods with fierce resource competition.

From the analysis presented in the previous section, the sampling period is not the only factor that affects the stability and the performance of each client in the NCS. The BU and the number of clients will also determine the time-delay and packet-loss levels of the NCS, which will affect the stability and the performance of each client. By (4), the sampling period and the BU are coupled parameters in the NCS. A large sampling period implies a smaller BU, thereby poor performance or even instability. A small sampling period implies a larger BU, more time delays, or even packet losses. Therefore, the trade-off between the sampling period and the stability is necessary to control the NCS effectively.

REFERENCES

[1] P. Ogren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Trans. on Automatic Control*, vol. 49, no. 8, pp. 1292-1302, August 2004.

[2] C. Meng, T. Wang, W. Chou, S. Luan, Y. Zhang, and Z. Tian, "Remote surgery case: Robot-assisted teleneurosurgery," *Proc. IEEE International Conference on Robotics and Automation*, vol. 1, pp. 819-823, May 2004.

[3] K. Hikichi, H. Morino, I. Arimoto, K. Sezaki, and Y. Yasuda, "The evaluation of delay jitter for haptics collaboration over the Internet," *Proc. IEEE Global Telecomm Conference*, vol. 2, pp. 1492-1496, November 2000.

[4] P. Seiler and R. Sengupta, "Analysis of communication losses in vehicle control problems," *Proc. 2001 American Control Conference*, vol. 2, pp. 1491-1496, June 2001.

[5] M. Björkbom, *Wireless Control System Simulation and Network Adaptive Control*, Ph.D. dissertation, Aalto University, Espoo, Finland, 2010.

[6] N. Boughanmi, H. Cruz-Sanchez, Y. Song, I. Al-Salami, and J. Yamé, "Case study of wireless networked control system: CONECS and COWNECS platform," *Proc. 2nd Workshop, 3SGS'09 CDROM*, November 2009.

[7] S. K. Cho, H. Z. Jin, J. M. Lee, and B. Yao, "Teleoperation of a mobile robot using a force-reflection joystick with sensing mechanism of rotating magnetic field," *IEEE/ASME Trans. on Mechatronics*, vol. 15, no. 1, pp. 17-26, February 2010.

[8] G. D. White, R. M. Bhatt, and C. P. Tang, "Experimental evaluation of dynamic redundancy resolution in a nonholonomic wheeled mobile manipulator," *IEEE/ASME Trans. on Mechatronics*, vol. 14, no. 3, pp. 349-357, June 2009.

[9] H. Mehrjerdi, M. Saad, and J. Ghommam, "Hierarchical fuzzy cooperative control and path following for a team of mobile robots," *IEEE/ ASME Trans. on Mechatronics*, vol. 16, no. 5, pp. 907-917, October 2011.

[10] Y. Wang, B. Kong, H. Leung, H. Jiang, and S. Chen, "An experimental vehicular wireless system and link performance analysis," *IEEE/ASME Trans. on Mechatronics*, vol. 17, no. 3, pp. 508-518, June 2012.

[11] D. Zhu, J. Guo, C. Cho, Y. Wang, and K.-M. Lee, "Wireless mobile sensor network for the system identification of a space frame bridge," *IEEE/ ASME Trans. on Mechatronics*, vol. 17, no. 3, pp. 499-507, June 2012.

[12] D. M. Anand, J. R. Moyne, and D. M. Tilbury, "Performance evaluation of wireless networks for factory automation applications," *Proc. 5th Annual IEEE Conference on Automation Science and Engineering*, pp. 340-346, August 2009.

[13] S. Dridi, O. Kallel, and S. Hasnaoui, "Performance analysis of wireless control area networks," *International Journal of Computer Science and Network Security*, vol. 7, no. 1, pp. 365-370, January 2007.

[14] M. Tabbara, D. Nešić, and A. R. Teel, "Stability of wireless and wireline networked control systems," *IEEE Trans. on Automatic Control*, vol. 52, no. 9, pp. 1615-1630, September 2007.

[15] V. Ungan, *Networked PID Controller for Wireless Systems*, M.S. thesis, KTH Royal Institute of Technology, Stockholm, Sweden. November 2010.

[16] Y. Zhang, X. Chen, and R. Lu, "Performance of networked control systems," *Mathematical Problems in Engineering*, vol. 2013, pp. 1-11, 2013.

[17] L. Zhang, H. Gao, and O. Kaynak, "Network-induced constraints on networked control systems—a survey," *IEEE Trans. on Industry Informatics*, vol. 9, no. 1, pp. 403-416, February 2013.

[18] Z.-H. Guan, C.-Y. Chen, G. Feng, and T. Li,

"Optimal tracking performance limitation of networked control systems with limited bandwidth and additive colored white Gaussian noise," *IEEE Trans. on Circuits and Systems I*, vol. 60, no. 1, pp. 189-198, January 2013.

[19] M. H. Lee, *Real-Time Networked Control with Multiple Clients*, M.S. thesis, Texas A&M University, College Station, TX, August 2009.

[20] P. Mantegazza, *DIAPM RTAI - Real-Time Application*, [Online] Available at http://www.rtai.org/

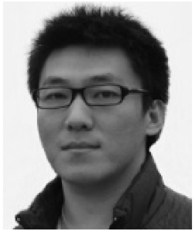
[21] Samba. [Online] Available at http://www.samba.org/

[22] N. J. Ploplys, P. A. Kawka, and A. G. Alleyne, "Closed-loop control over wireless network," *IEEE Control System Magazine*, vol. 24, no. 3, pp. 57-71, June 2004.

[23] I. Lopez, J. L. Piovesan, C. T. Abdallah, D. Lee, O. Martinez, and M. Spong, "Practical issues in networked control systems," *Proc. 2006 American Control Conference*, pp. 4201-4206, June 2006.

[24] K. Ji and W.-J. Kim, "Optimal bandwidth allocation and QoS-adaptive control co-design for networked control systems," *International Journal of Control, Automation, and Systems*, vol. 6, no. 4, pp. 596-606, August 2008.

[25] J. W. S. Liu, *Real-Time Systems*, Prentice Hall, Upper Saddle River, NJ, 2000.



Jiawei Dong received his B.S. degree in Mechanical Manufacture and Automation from the Beijing Technology and Business University (BTBU), Beijing, China in 2005, an M.S. degree in Advanced Robotics from the Beijing University of Aeronautics and Astronautics (BUAA), Beijing, China in 2008, and a Ph.D. degree in Mechanical Engineering, Texas A&M University (TAMU), College Station, TX in 2013. His research interests include networked control systems, real-time systems, optimization, and advanced robotics.

Won-jong Kim received his B.S. (*summa cum laude*) and M.S. degrees in Control and Instrumentation Engineering from Seoul National University, Seoul, Korea, in 1989 and 1991, respectively, and his Ph.D. degree in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology (MIT), Cambridge, in 1997. Since 2000, he has been with the Department of Mechanical Engineering, Texas A&M University (TAMU), College Station, where currently he is Associate Professor and was the inaugural Holder of the Gulf Oil/Thomas A. Dietz Career Development Professorship II in 2007-2010. His current research interests include the analysis, design, and real-time control of mechatronic systems, networked control systems, and nanoscale engineering and technology. He is a holder of three U.S. patents on precision positioning systems. He is Fellow of ASME, Senior Member of IEEE, and Member of Pi Tau Sigma. Prof. Kim is an Associate Editor of *ASME Journal of Dynamic Systems, Measurement and Control, International Journal of Control, Automation, and Systems*, and *Asian Journal of Control*, and was Technical Editor of *IEEE/ASME Transactions on Mechatronics*.