Effects of Local and Global Adaptive Δ -Causality Control on Cooperative Work between Remote Robot Systems with Force Feedback

Nuzrath Hameedha A[†], Yutaka Ishibashi (member)[†]

Abstract In this paper, we investigate effects of two kinds of the adaptive Δ -causality control, which adjusts the output timing of the position information among multiple terminals, on cooperative work between two remote robot control systems with force feedback by experiment. One is the local control under which the adaptive Δ -causality control is partially applied to the systems, and the other is the global control under which the adaptive Δ -causality control is globally done. In each system, a user remotely operates a robot having a force sensor by using a haptic interface device while watching video. We conduct cooperative work of carrying an object grasped by two robot arms and compare the two kinds of control in our experiment. Experimental results illustrate that the global control is more effective than the local control for the cooperative work.

Key words: Remote robot system, force feedback, cooperative work, QoS control, network delay, experiment

1. Introduction

Remote robot control has been actively researched in the recent years. Remote robot systems with force feedback find their applications from manufacturing units in factories to intensive care units in hospitals for procedures like tele-surgery¹⁾⁻⁷⁾. In such a system with force feedback, a user can remotely operate a robot with a force sensor by using a haptic interface device while watching video. With the force feedback, the user can perceive the force exerted by the remotely-located robot. The efficiency and accuracy of the system can be enhanced by using multiple remote robot systems with force feedback⁸⁾. When the force information is transmitted through a network like the Internet, which does not guarantee the quality of service (QoS)⁹⁾, there may be serious degradation of QoS and QoE (Quality of Experience)¹⁰⁾ due to the network delay, delay jitter, and packet loss. The operability of the system can largely be affected by the instability caused by the unfavorable conditions. To achieve high QoS, it is crucial to implement QoS control⁶⁾ and stabilization control⁷⁾.

In a previous study⁸⁾, the authors investigate the influence of network delay in cooperative work of carrying an object between two remote robot systems with force

feedback and find that the force exerted on the object increases as the network delay becomes larger. They also show that larger force is applied while changing the direction of the movement during the cooperative work. If the object is fragile, the large force may break the object. Therefore, we need to avoid large force. In another previous study¹¹⁾, the authors apply the adaptive Δ -causality control¹²⁾, which adjusts the output timing of position information dynamically according to the network delay, to the two robots to reduce the force exerted on the object. They demonstrate the effectiveness of the control by experiment. However, influences of network delay between robots, and network delay between the haptic interface device and robot have not been clarified. Also, the work is conducted by a single user; that is, the user manipulates two haptic interface devices to move the remote robots by both hands. Since the devices can be manipulated by two users, we need to clarify the effect of the adaptive Δ -causality control for the cooperative work by two users. Furthermore, the control can be applied to the robot and haptic interface device of each system. However, the effect of the control in this case has not been investigated so far.

In this paper, we propose the local and global adaptive Δ -causality control to mitigate influences of network delay for the cooperative work of carrying an object together for two remote robot systems with force feedback. In the local adaptive Δ -causality control, the adaptive Δ -causality control is carried out between two

Received May 27, 2021; Revised August 24, 2021; Accepted September 24, 2021

[†] Nagoya Institute of Technology
(Nagoya, Japan)

the slave terminal by UDP (User Datagram Protocol)¹⁷⁾ as the transport protocol. The slave terminal employs the real-time control function¹⁶⁾ to acquire the information about the position of the robot arm, and the terminal uses the real-time monitor function¹⁶⁾ to get the information about the force sensor from the robot controller every 3.5 milliseconds⁵⁾. The two types of information are transmitted from the robot controller to the slave terminal by UDP. The slave terminal sends the information to the master terminal in each system.

The slave terminal of each system sends the position information of the robot arm to that of the other system, to judge the direction of movement of the robot arms during the cooperative work. Each slave terminal judges in this paper that the movement direction is changed when each robot arm has continuously moved in the opposite direction 50¹¹⁾ or more times.

In our experiment, two different users manipulate the haptic interface devices of System 1 and System 2 while watching videos.

3. Local and Global Adaptive Δ -Causality Control

In the adaptive Δ -causality control¹¹⁾, the output timing of the position information is delayed dynamically according to the network delay. The output timing of the position information is set to the generation time (i.e., timestamp) + Δ (>0) seconds, and the value of Δ is changed dynamically according to the network delay. The time when the position information is generated at a source (i.e., each master or slave terminal in this paper) is attached as the timestamp along with the position information that is transmitted to destinations (each master or slave terminal). We here assume that the global clock¹¹⁾ is used; that is, the clock ticks have the same value and speed among the systems. Each terminal should send the current value of Δ to the other terminal at regular intervals as well as when the value is changed owing to network delay jitter¹¹⁾ (we do not handle the delay jitter for simplicity in this paper as mentioned in Section 4; note that we can absorb the delay jitter to some extent by setting the value of Δ as the network delay plus the buffering time for the jitter¹²⁾). The terminal selects the largest value as Δ from among the latest-received values (including its own value) for simplicity in this paper¹¹⁾. All the terminals use the same method to determine the value of Δ . Let us denote the value of Δ at time t ($t \geq 1$) by Δ_t here. The

value of Δ_t is obtained by smoothing the network delay d_t measured at time t as follows:

$$\Delta_1 = d_1 \quad (1)$$

$$\Delta_t = \alpha \Delta_{t-1} + (1-\alpha)d_t \quad (t \geq 2) \quad (2)$$

where α is a smoothing coefficient and is set to 0.998¹¹⁾.

In this paper, two kinds of the adaptive Δ causality control are handled. One is the local adaptive Δ -causality control (called *LADC* here), and the other is the global adaptive Δ -causality control (called *GADC*). In LADC, we have two cases: Control between the two robots (*LADC-RR*), and control between the haptic interface device and robot in each system (*LADC-DR*). In GADC, both LADC-RR and LADC-DR are carried out in the systems.

3.1 LADC

(1) LADC-RR

When the network delay between the two robots increases, the difference in position between the two robots becomes larger. This difference in position results in large force applied to an object carried by the robots especially when the direction of movement is changed. In LADC-RR, the output timing of the position information of each robot is delayed dynamically according to the network delay between the robots so that both the robots move at the same time. In this paper, for simplicity, the network delay from the industrial robot of System 1 (referred to as *robot 1*) to the industrial robot of System 2 (*robot 2*) is set to the same as that from robot 2 to robot 1.

(2) LADC-DR

In this case, the adaptive Δ causality control is applied to reduce the influence of difference in network delay between the haptic interface device and robot of each system between the two systems. The value of Δ in each system are set so that each robot is moved at

Fig. 2: Example of camera view.

the same time as the other robot. For instance, when the delay between the haptic interface device and robot in System 1 is 0 ms and the delay in System 2 is 100 ms, the value of Δ is set to 100 ms in this case so as to decrease position difference between the two robots.

3.2 GADC

GADC is a combination of LADC-RR and LADC-DR. In GADC, the output timing of each robot is adjusted for both network delays between the robots and the difference in network delay between the two systems rather than only one of them as in the cases of LADC-RR and LADC-DR. For instance, let us set the network delay between the haptic interface device and robot in System 1 to 0 ms, the network delay in System 2 to 100 ms, and the network delay between the robots to 100 ms. In GADC, the values of Δ at the master and slave terminals of each system are set to 100 ms because the largest value of network delays is selected as Δ as described earlier. Note that in LADC-RR, Δ at the slave terminal of each system is set to 100 ms; in LADC-DR, Δ of the master terminal of each system is set to 100 ms.

4. Experiment Method

In our experiment, we carried out cooperative work of carrying a wooden stick as the object grasped by the two robot arms¹¹⁾. In this paper, two different users operated the two haptic interface devices with their dominant hands while perceiving the force and watching videos.

The wooden building blocks were piled-up before and behind the wooden stick, and a paper block was placed on each uppermost building block as shown in Fig. 2. The building blocks are arranged so that the two paper blocks are at the same height. The task was to touch each paper block (front and back) by the wooden stick. The two paper blocks were placed at 80 mm from each other. The initial position of the wooden stick was set so that it is at an equal distance from both the paper blocks (i.e., 40 mm from each paper block). We touched the paper block on the front side in about 5 seconds and that on the back side in around 10 seconds (i.e., it took about 15 seconds to complete the task). This method is used to move the wooden stick in the same way so that we can maintain almost the same movement throughout the experimentation. The two users started each task at almost the same time by hearing a voice cue.

The force mapping ratio between the haptic interface device and the robot arm in each system was set to

1:3¹⁴⁾. The robot arm of each system was allowed to move only in front and back direction (i.e., the x -axis) (see Fig. 2); the movement in left and right (the y -axis), and up and down (the z -axis) directions was restricted for simplicity.

The two remote robot systems (i.e., Systems 1 and 2) are connected through a network emulator (NIST Net¹⁸⁾) instead of the network in Fig. 1 to add a constant delay to each packet sent between the two robots, and between the master and slave terminals of each system (the one-way constant delay is called the *additional delay* here); the constant delays in both directions are assumed to be same. The additional delay in System 1 (i.e., that between the master and slave terminals of System 1), that in System 2, and that between the two robots are represented as *additional delay 1*, *additional delay 2*, and *additional delay 3*, respectively (see Fig. 1). We do not produce any packet loss by the network emulator, for simplicity. LADC-RR is used to alleviate the influences of additional delay 3, LADC-DR to alleviate the influences of additional delays 1 and 2, and GADC for all the three additional delays. The combination of the delays is expressed as (additional delay 1, additional delay 2, additional delay 3) in this paper. The experiment was carried out for different additional delays between the two slave terminals, and between the master and slave terminals of each system. Additional delays 1 and 2 were changed from 0 ms to 200 ms at intervals of 100 ms. Additional delay 3 was varied between 0 ms to 100 ms. The work was repeated 10 times for each combination of additional delays. To clarify effects of LADC-RR, LADC-DR, and GADC, we also handle the case where the adaptive Δ -causality control is not carried out (called *No control* here). We use the average of average absolute force and the average of maximum absolute force at each robot as performance measures. The average of the average absolute force at each robot is defined as the 10 times average of the temporal average absolute force at the robot. The average of maximum absolute force is obtained by averaging the maximum absolute force for all the tasks at the robot.

5. Experimental Results and Considerations

Fig. 3 shows the average of average absolute force and the average of maximum absolute force for No control and LADC-RR, and Fig. 4 plots those for No control and LADC-DR. Also, Fig. 5 shows those for LADC-RR,

(a) Average of average absolute force

(b) Average of maximum absolute force

Fig. 3: Comparison between No control and LADC-RR.

LADC-DR, and GADC. The 95% confidence intervals of averages are also plotted in the figures.

We see from Fig. 3 that the average of average absolute force and the average of the maximum absolute force of LADC-RR are smaller than those of No control. We also observe that the differences between LADC-RR and No control when additional delay 3 is 100 ms are larger than those when additional delay 3 is 0 ms. This is the effect of LADC-RR.

On the other hand, from Fig. 4, we find that the differences between No control and LADC-DR are not so large, compared with those between No control and LADC-RR (Fig. 3). Therefore, we can infer that LADC-RR has a larger effect than LADC-DR.

From Fig. 5, we see that the average of average absolute force and the average of maximum absolute force are the smallest for GADC. Thus, we can say that GADC is more effective than LADC-RR and LADC-DR. It should be noted that GADC may damage the interactivity, but GADC is better than LADC for the network delays considered in this paper. Fig. 5 also reveals that the differences between GADC and LADC-RR are not large. Thus, we carried out t-test to examine whether the differences are significant. As a result of t-test, we found that there exist significant differences between GADC and LADC-RR. The t-test result between averages of average absolute force for robot 1 was as follows: $[t(9) = 4.26, p = 0.0013, \text{one-sided test}]$;

(a) Average of average absolute force

(b) Average of maximum absolute force

Fig. 4: Comparison between No control and LADC-DR.

for the other combinations, the t values were larger than 4.26 and the p values were less than 0.0013; this means that GADC is more effective than LADC-RR.

From Figs. 3, 4, and 5, we notice that the average of average absolute force and average of maximum absolute force of robot 1 tend to be smaller than those of robot 2. This is because the two systems are operated by two different users; we confirmed that the difference reversed, when the users switched the robots. We also found that the direction of force of robot 1 is opposite to that of robot 2 (this will be seen later).

To examine the differences among LADC-RR, LADC-DR, and GADC in Fig. 5 in further detail, we show the force versus the elapsed time from the beginning of task for LADC-RR, LADC-DR, and GADC for delay combination (0,100,100) ms in Fig. 6. The results are typical examples of force versus the elapsed time for LADC-RR, LADC-DR, and GADC in our experiment. From Fig. 6, we can clearly see that the force is large for LADC-DR. The force fluctuation is also large especially during the change in the movement direction at about 6 seconds. On the other hand, the force for LADC-RR and GADC hardly increases, and the force exerted during the direction change is much smaller than that of LADC-DR. By comparing Figs. 6 (a) and (c), we see that GADC has smaller force than LADC-RR.

(a) Average of average absolute force

(b) Average of maximum absolute force

Fig. 5: Comparison among LADC-RR, LADC-DR, and GADC.

In addition, we conducted the experiment for other delay combinations such as (0,200,100) ms and (100,200,100) ms, and we obtained almost the same results as those in this paper. From these considerations, we can conclude that GADC is more effective than LADC.

6. Conclusion

In this paper, we investigated the effects of the global and local adaptive Δ -causality control on cooperative work between the two remote robot systems with force feedback. As a result, we found that the global adaptive Δ -causality control is more effective than the local adaptive Δ -causality control by experimentation. We also confirmed that there exist significant differences between results for the local adaptive Δ -causality control between robots and the global adaptive Δ -causality control.

As the next step of our research, we will focus on collaboration between two users (i.e., the local adaptive Δ -causality control between the two haptic interface devices) to achieve the most effective QoS control. We will also examine effects of the global and local adaptive Δ -causality control for a variety of network environments and system parameters. We also plan to reduce the

Fig. 6: Force versus elapsed time for delay combination (0,100,100) ms.

force fluctuation during the direction change and to apply the global adaptive Δ -causality control to different types of cooperative work.

Acknowledgement

The authors thank Mr. Ye Ruzhou for his help with experimentation.

References

- 1) P. Nadrag, L. Temzi, H. Arioui, and P. Hoppenot: "Remote control of an assistive robot using force feedback," *Proceedings of The 15th International conference on Advanced Robotics*. pp. 211-216 (June 2011)

- 2) E. Steinbach, S. Hirche, M. Emst, F. Brandi, R. Chaudhari, J. Kammerl, and I. Vittorias: "Haptic communications," *IEEE Journals & Magazines*. 100 4, pp. 937-945 (April 2012)
- 3) N. Motoi and S. Nakamura: "Remote Control Method with Tactile Sensation for Underwater Robot with Magnetic Coupling," *IEEE Advanced Motion Control*. pp. 357-362 (September 2020)
- 4) T. Miyoshi, Y. Maeda, Y. Morita, Y. Ishibashi, and K. Terashima: "Development of haptic network game based on multi-lateral tele-control theory and influence of network delay on QoE," (in Japanese), *VRSJ*. 19 4, pp. 559-569 (December 2014)
- 5) K. Suzuki, Y. Maeda, Y. Ishibashi, and N. Fukushima: "Improvement of operability in remote robot control with force feedback," *IEEE GCCE*. pp. 16 -20. (October 2015)
- 6) Y. Ishibashi and P. Huang: "Improvement of QoS in haptic communication and its future," (in Japanese), *IEICE Trans. Commun.* J99-B 10, pp. 16-20 (October 2016)
- 7) T. Miyoshi, K. Terashima, and M. Suss: "A design method of wave filter for stabilizing non-passive operating system," *IEEE ICCA*. pp. 1318-1324 (October 2006)
- 8) E. Taguchi, Y. Ishibashi, P. Huang, and Y. Tateiwa: "Experiment on collaborative work between remote robot systems with haptics," (in Japanese), *IEICE General Conference*. B-11-17 (March 2018)
- 9) ITU-T Rec. I. 350: "General aspects of quality of service and network performance in digital networks," (1993).
- 10) ITU-T Rec.G. 100/P. 10 Amendment 1: "New appendix I –Definition of quality of experience (QoE)," (2007).
- 11) K. Kanaishi, Y. Ishibashi, P. Huang, and Y. Tateiwa: "Effect of adaptive Δ -causality control for cooperation between remote robot systems with force feedback: Case where robots have equal relationship," *IEICE Technical Report*. NS2019-119 (October 2019)
- 12) Y. Kusunose, Y. Ishibashi, N. Fukushima, and S. Sugawara: "Adaptive delta-causality control with prediction in networked real-time game using haptic media," *Proceedings of Asia-Pacific Conference on Communications*. pp. 800-805 (October 2012)
- 13) E. Taguchi, Y. Ishibashi, P. Huang, Y. Tateiwa, and T. Miyoshi: "Comparison of stabilization control in cooperation between remote robot systems with force feedback," *IJMERR*. 9 1, pp. 87-92 (January 2020)
- 14) P. Huang, T. Miyoshi, and Y. Ishibashi: "Enhancement of stabilization control in remote robot system with force feedback," *IJCNS*. 12 7, pp. 99-111 (July 2019)
- 15) 3D Systems Touch: <https://www.3dsystems.cpm/haptic-devices/touch>.
- 16) RV-2F-D Series: Standard Specifications, <http://www.mitsubishielectric.co.jp/dl/fa/members/document/manual/robot/bfp-a8899/bfp-a8899u.pdf>.
- 17) IETF: RFC768 "User Datagram Protocol," (August 1980)
- 18) M. Carson and D. Santay: "NIST Net: a Linux-based network emulation tool," *ACM SIGCOMM*. 33 3, pp. 111-126 (July 2003)

Nuzrath Hameedha A received the B.E. and M.E. degrees from Anna University, India, in 2010 and 2013, respectively. She is currently working towards her Ph.D. in Department of Computer Science, Graduate School of Engineering, Nagoya Institute of Technology. Her research interests include QoS (Quality of Service) control and QoE (Quality of Experience) assessment in remote robot systems with force feedback. She is a Student Member of IEEE.

Yutaka Ishibashi received the B.E., M.E., and Dr.E. degrees from Nagoya Institute of Technology, Japan in 1981, 1983, and 1990, respectively. In 1983, he joined Musashino Electrical Communication Laboratory of NTT. From 1993 to 2001, he served as an Associate Professor of Department of Electrical and Computer Engineering, Faculty of Engineering, Nagoya Institute of Technology. Currently, he is a Professor of Graduate School of Engineering, Nagoya Institute of Technology. His research interests include networked multimedia, QoS control, and remote robot control with force feedback. He is a Fellow of IEICE, a Senior Member of IEEE, and a Member of ACM, IPSJ, VRSJ, and IEEJ.