Remote Robot Control with Low-cost Robotic Arms and Human Motions

Jordan Fogg*
jordan.fogg@okstate.edu
Oklahoma State University
Stillwater, Oklahoma, USA

Zeyu Deng*
zdeng6@lsu.edu
Louisiana State University
Baton Rouge, Louisiana, USA

Emory Meursing
emeursi@ostatemail.okstate.edu
Oklahoma State University
Stillwater, Oklahoma, USA

Long Huang
lhuan45@lsu.edu
Louisiana State University
Baton Rouge, Louisiana, USA

Huaxia Wang
huaxia.wang@okstate.edu
Oklahoma State University
Stillwater, Oklahoma, USA

Chen Wang
chenwang1@lsu.edu
Louisiana State University
Baton Rouge, Louisiana, USA

ABSTRACT

Geographically-separated people are now connected by smart devices and networks to enjoy remote human interactions. However, current online interactions are still confined in a virtual space. Extending the pure virtual interactions to the physical world requires multidisciplinary research efforts, including sensing, robot control, networking, and kinematics mapping. This paper introduces a remote motion-controlled robotic arm framework by integrating these techniques, which allows a user to control a far-end robotic arm simply by hand motions. In the meanwhile, the robotic arm follows the user's hand to perform tasks and sends back its live states to the user in video stream. Furthermore, we explore using cheap robotic arms and off-the-shelf motion capture devices to facilitate the wide use of the platform in people's daily life. No professional knowledge is required from the user. Moreover, we implement a testbed that connects two US states for the remote control study. We investigate different types of latency that affect the user's remote control experience and conduct comparative studies. Results show that the current commercial motion capture device, low-cost robotic arms and networks are already available to provide physically-augmented remote human interactions.

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous computing**; • **Computer systems organization** → *Robotic control*; *Embedded and cyber-physical systems.*

KEYWORDS

Human-robot Interaction; Teleoperation

ACM Reference Format:
Jordan Fogg, Zeyu Deng, Emory Meursing, Long Huang, Huaxia Wang, and Chen Wang. 2022. Remote Robot Control with Low-cost Robotic Arms and Human Motions. In *Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp/ISWC '22 Adjunct), September 11–15, 2022, Cambridge, United Kingdom.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3544793.3560331

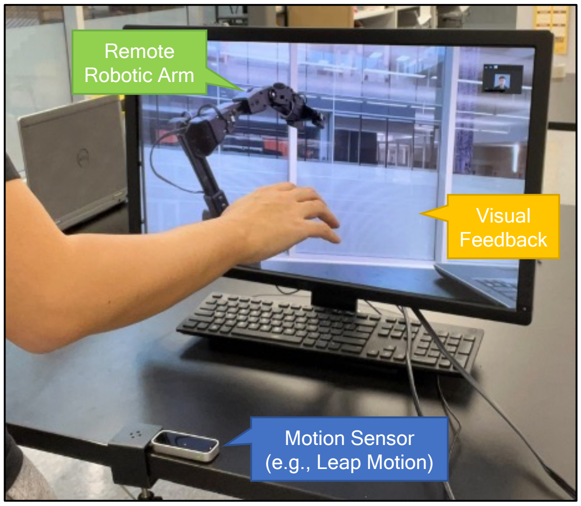

Figure 1: Illustration of remote control.

1 INTRODUCTION

Smart devices connect geographically separated people through the network and enable a multitude of daily activities to go remote, including telecommuting, social networks, online shopping, web conferences, virtual classrooms, and remote diagnoses. The outbreak of the COVID-19 pandemic further boosts such a "Go Remote" trend when the face-to-face social activities might be under a high health risk. However, these remote human activities are still constrained in the virtual space with limited applications. There is an imperative need for connecting virtual activities to the real world.

Remote real-time motion tracking has found a growing interest in robotics for robot imitation control [1, 4]. Motion retargeting is the essential and challenging part of real-time robot control from human observations since humans and robots are dissimilar in size, degrees-of-freedom (DOF), and dynamics and mechanical limitations [3]. However, prior research on teleoperation systems require to consider the hardware of very expensive robotic arm (e.g.,

---

[*] Both authors contributed equally to this research.

UbiComp/ISWC '22 Adjunct, September 11–15, 2022, Cambridge, United Kingdom

ACM ISBN 978-1-4503-9423-9/22/09.
https://doi.org/10.1145/3544793.3560331

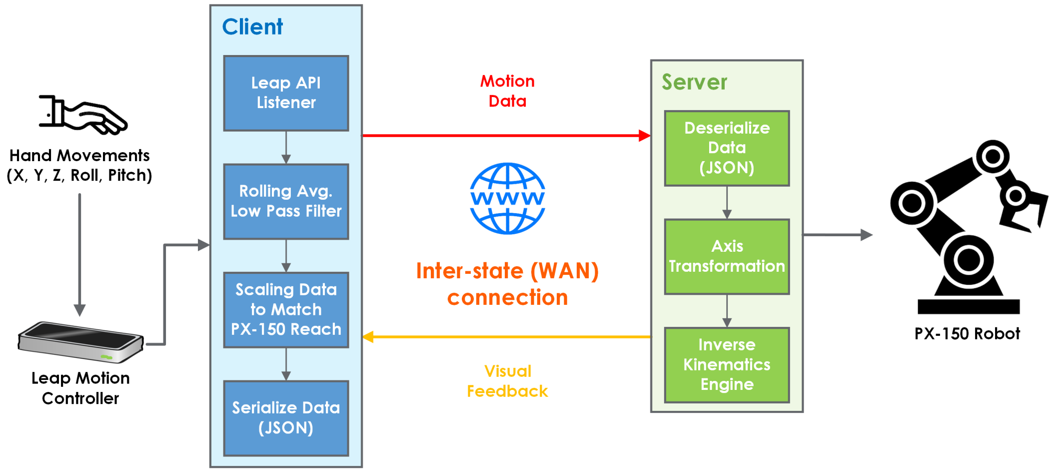

Figure 2: The remote-controlled robotic arm platform.

$20, 000), which implies overwhelming high costs of development and makes it unaffordable to home use.

In this paper, we aim at bringing cheap robotic arms to home to enable physically-augmented online interactions between people. We introduce a remote motion-controlled robotic arm framework as illustrated in Figure 1, which allows users to control a remote-end robotic arm simply by hand motions in real-time. Simultaneously, the robotic arm executes the network transmitted commands to follow the user's hand and return visual feedback via video streams. Thus, a control loop is formed between the user and the far-end robotic arm for interactive control. However, a number of challenges need to be addressed when the robotic arm control goes interactive and remote. In particular, when a user controls a robotic arm with hand motions, the sampling errors of the motion capture device and the varying speeds of the human arm would cause the robotic arm movement to be noisy and even easy to fail. Moreover, the network traffic delays and transmission latency would cause the far-end robot hard to perform real-time control tasks.

Our contributions are summarized as the following:

- We develop a robotic arm framework using a low-cost robotic arm and the off-the-shelf motion capture device to enable remote real-time control over a Wide Area Network (WAN). It shows the potential to provide physically-augmented services to users with low cost and nonprofessional knowledge.
- A testbed bridging two US states (i.e., Oklahoma and Louisiana with a distance over 600 miles) is built for estimating the availability of the remote motion-controlled robot.
- We examine different types of latency that impact the user experience and task performance. Moreover, we conduct a comparison study between remote control and local replay.

2 SYSTEM DESIGN

The main goal of this paper is to develop a motion-controlled robotic arm platform using commodity devices and existing networks for achieving physically-augmented remote human interaction. Figure 2 shows the low-cost platform built upon two ends (i.e., client and server) connected by a WAN. The client captures the human hand movements using a Leap Motion Controller and sends the processed position data to the server via the WAN. The server then transforms the received data into command sequences and publishes them to a PincherX 150 Robot Arm for task execution, whose visual feedback is returned to the client via video streams. The PCs at both ends are installed with Linux Ubuntu 18.04.

Client Design. By accessing the LEAP Motion API, the client starts a listener that takes in the tracking data from the sensor. When a hand is in view, the program takes the rolling average of the palm center's positions within the most recent three frames in order to get an accurate position value for the robotic arm to use. PX-150 has a work space that "a specification of the configurations that the end-effector of the robot can reach" [2]. In order for the PX-150 to configure into a correct position, we scale the position values from the LEAP Motion Controller (e.g., x, y, and z) into position values that can be accepted by the PX-150 (e.g., $\hat{x}$, $\hat{y}$, and $\hat{z}$). The processed data are written to a list and serialized into a JSON string to be sent to the server over the web socket.

Server Design. Once the server has received the JSON string, it deserializes the string back into a list. The list is then published to the kinematics engine using the *set-ee-pose-components* method, which is part of the arm library of the PX-150 that houses all of the kinematics calculation data. If a successful path was found, it will publish a list of angles for each of the servos to manipulate the arm to achieve the desired end-effector position. If a path is not found, it will return that the robot has failed to converge to the specified position and will stay in the last position it reached. The movements will continue so long as there maintains an active connection between the server and the client computers.

3 PRELIMINARY RESULTS

We first conduct a statistical study about the control delay of our system under four different connection configurations: 1) straight connection (a connection with no web socket), 2) LAN connection

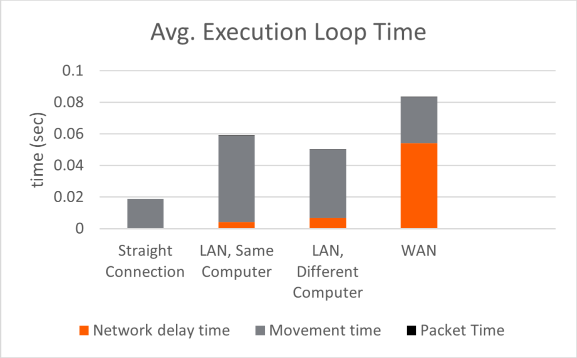

Figure 3: Average time to execute one iteration of the program.

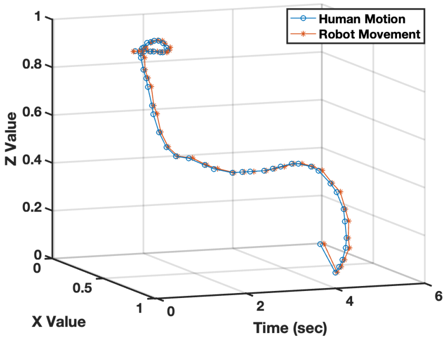

Figure 4: Latency between the robot and human in remote control.

on the same computer, 3) LAN connection on different computers, and 4) WAN connection across multiple states using a VPN. As shown in Figure 3, the execution time under the straight connection and two LAN connections is mainly composed of the robot's movement time, which is around 0.018s, 0.05s, and 0.04s, respectively. However, the main component of the execution time under the WAN connection is the network delay, whose average is 0.052s, and the robot's movement time is around 0.023s. In regard to the total execution time, the straight connection achieves the shortest time around 0.02s, the two LAN connections show a longer time between 0.05s and 0.06s, and the WAN connection turns to have the longest time around 0.08s.

We next compare the motions of the human hand and the remote-controlled robot to study the effects of transmission and robot execution on the motion trajectories. As we can observe in Figure 4, when drawing a "S", the robot trajectory is close to the human hand trajectory, which indicates that the robot follows the human movement well and the latency during the remote control has a very slight impact on robot trajectory. On the other hand, although the axes values are not differentiated, we can observe the robot movement over time is not as smooth as human motion, and the time shift is not constant for every frame, indicating the robot executing speed is swinging in a small range.

4 CONCLUSION AND FUTURE WORK

We developed a low-cost robotic arm platform with an off-the-shelf motion capture device to enable remote real-time control over a WAN. We studied its control delay performance under different LAN/WAN connection configurations, and observed that the dominant part of execution time for the WAN connection is the network delay instead of the robot's movement time. Experiments show that the current cheap robotic arms and motion sensors are capable of supporting physically-augmented remote human interactions. In the future, we plan to further reduce the system's execution time by speeding up the robot's movements and decreasing the overall delays in the system. The current plan involves: 1) adopting the low-latency 5G networks to deploy the robotic arm platform, 2) implementing a queue system in order for one thread to put the position data received from the client in the queue and utilize another thread to retrieve data from the queue with a speed-up strategy to execute movements on the robot at the cost of slight resolution loss, and 3) constructing a 3D robot simulator for the client to reduce video streaming delay.

ACKNOWLEDGMENTS

This work was partially supported by LABoR LEQSF(2020-23)-RD-A-11 and NSF CNS-2155131.

REFERENCES

[1] Miguel Arduengo, Ana Arduengo, Adrià Colomé, Joan Lobo-Prat, and Carme Torras. 2019. A Robot Teleoperation Framework for Human Motion Transfer. *arXiv preprint arXiv:1909.06278* (2019).
[2] Kevin M Lynch and Frank C Park. 2017. *Modern robotics.* Cambridge University Press.
[3] Guilherme Maeda, Marco Ewerton, Dorothea Koert, and Jan Peters. 2016. Acquiring and generalizing the embodiment mapping from human observations to robot skills. *IEEE Robotics and Automation Letters* 1, 2 (2016), 784–791.
[4] Yongsheng Ou, Jianbing Hu, Zhiyang Wang, Yiqun Fu, Xinyu Wu, and Xiaoyun Li. 2015. A real-time human imitation system using kinect. *International Journal of Social Robotics* 7, 5 (2015), 587–600.