

THE JACONTROL: DEVELOPMENT OF A SMARTPHONE INTERFACE FOR THE ASSISTIVE ROBOTIC MANIPULATOR

Cheng-Shiu Chung, MS^{1,2}, Joseph Boninger^{1,3}, Hongwu Wang, PhD^{1,2}, Rory A. Cooper, PhD^{1,2}

¹*Human Engineering Research Laboratories, Department of Veterans Affairs, Pittsburgh, PA;*

²*Department of Rehabilitation Science and Technology, University of Pittsburgh, Pittsburgh, PA;*

³*Swarthmore College, Swarthmore, PA*

ABSTRACT

Assistive Robotic Manipulators (ARMs) are an assistive technology with great possible benefits for users with disabilities that limit the functioning of their hands or arms. However, the current commercial physical ARM controllers, keypad and 3-axis joystick may be difficult to use for persons with severe muscle weakness, paralysis, or an impaired ability to grip. Moreover, their wiring cable may limit the flexibility in installation and operation. To overcome these challenges, we have developed the Jacontrol, a framework for the Jaco ARM that has the enhanced capabilities for robotic autonomy and employs a customizable graphical user interface (GUI).

We developed two versions of the Jacontrol GUIs, touch-joystick and touch-keypad, and compared their performance on the standardized ADL task board against the Jaco's default joystick. Our results suggest the Jacontrol is a viable and possibly even easier and more efficient to the joystick.

BACKGROUND

Assistive Robotic Manipulators (ARMs) have great potential to increase functionality and independence for users with disabilities in the upper extremities (Bach, Zeelenberg, & Winter, 1990; C.-S. Chung, Wang, & Cooper, 2013b). ARMs can enable users to grasp objects that would otherwise be out of their reach and perform activities of daily living (ADLs) (Wang et al., 2013). Numerous ARMs have become commercially available in recent years, and one of the most recent is the Jaco ARM developed by Kinova Technology. The Jaco ARM has six interlocking joints, three "fingers" for gripping, a reach of ninety centimeters, and can move through six degrees of freedom.

The ARM Control Interfaces

There are two primary physical control interfaces from the commercial ARMs: custom joystick and keypad. The joystick has three degrees of freedom—horizontal movements including left/right and forward/backward and twisting movement such as clockwise/counter-clockwise. Two push buttons on the joystick knob allow the user to toggle between three common control modes—translation mode, in which joystick movements change the position of

the ARM's hand; wrist rotation mode, in which joystick movements change the orientation of the hand; and finger mode, in which joystick movements open and close three fingers). The joystick is simple and efficient, but changing modes and the twisting motion may be difficult or even impossible for some users to manipulate (C. Chung, Hannan, Wang, Kelleher, & Cooper, 2014).

Alternatively, keypad control interface replaces each direction of motion with keys—each key corresponds to only one single motion such as moving left. Multiple clicking on the same key accelerates the ARM movement and clicking on the key with the opposite direction decelerates or reverses the ARM motion. The keypad is more affordable for users with difficulties using joystick, but memorizing key functions and hierarchy may be demanding (Tsui, Kim, Behal, Kontak, & Yanco, 2011).

The aforementioned control interfaces can only be operated with a cable linked to the ARM. This limits the flexibility in ARM operation once the user is transferred to other locations such as bed. A developing remote interface using Microsoft Kinect gesture or voice recognition (Jiang, Zhang, Wachs, & Duerstock, 2014) was tested with pick-and-place, drinking, and photo shooting tasks. However, the modification to mount the Kinect in front of the chair may enlarge the footprint and increase difficulties for indoor maneuverability. In addition, it has not yet been verified with different environments or outdoor conditions.

The smartphone's multi-touch interface and the built-in Bluetooth and Wi-Fi connectivity have the potential to overcome the limitations of current control interfaces and provide wire-free manipulation assistance. Statistic reveals that smartphone users in America are increasing dramatically in recent years. Moreover, 83% of users keep the phone turned on and within reachable range (Dicianno et al., 2014; Wu, Liu, Brown, Kelleher, & Cooper, 2013).

Therefore, the purpose of the current development is to establish a framework that provides customizable smartphone user interfaces and to test their feasibility and performance. We have developed a unique, assistive platform for the Jaco ARM called the Jacontrol, which can mimic the functionality of the joystick and keypad through a touchscreen graphical user interface (GUI) and has the expendability for autonomous robotic functionalities.

METHODOLOGY

The Jacontrol Framework

Our platform for the Jaco ARM, the Jacontrol, is composed of two components: a smartphone application on an Android device and a program on a Linux computer (Figure 1). The smartphone app matches the capabilities of the joystick and keypad, but requires minimal physical exertion and no difficult motions such as grasping or twisting. The Linux program communicates with smartphone app through Bluetooth or Wi-Fi and autonomous functions and object recognitions developed under Robotic Operating System (ROS) by (C.-S. Chung, Wang, & Cooper, 2013a) and eventually control the ARM through the LibKinDrv Driver (<http://www.fawkesrobotics.org/projects/libkindrv/>). Presently, we use a laptop as the Linux computer for testing the feasibility. However, it can be replaced with small-sized single board computer

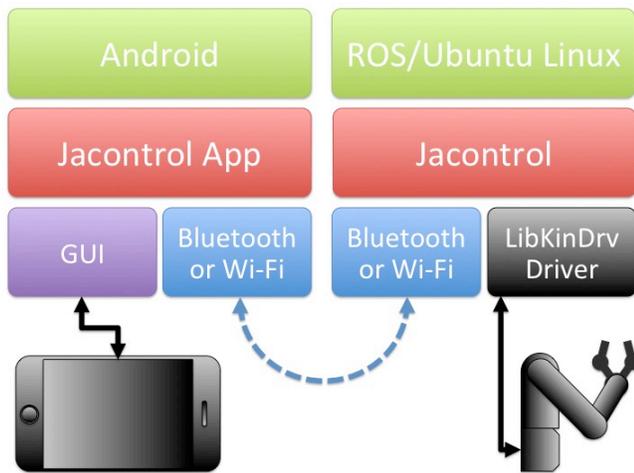


Figure 1: The framework of Jacontrol

Graphical User Interfaces for the Jacontrol App

We have developed two GUIs to mimic the two primary ARM control interfaces (Figure 2). The touch-keypad GUI (Figure 2 left) was built to mimic the interface developed for the Personal Mobility and Manipulation Appliance (PerMMA) (Wang et al., 2013). Touch-keypad uses categorized buttons—each button corresponds to an ARM motion, and pressing a button multiple times in a row causes the ARM to accelerate the motion. The two exceptions to this are the “Close Hand” and “Open Hand” buttons, which must be held down and can only move the ARM fingers at one speed. This GUI design with text descriptions not only facilitates to quickly identify the function of ARM motions, but also reduce the frustration in memorizing key functions. In addition, clicking anywhere on the GUI’s blue background stops all ARM motions. This shorter distance from the motion key button to the stop reduces the time in finding the stop key and furthermore, improves ARM accuracy.

Figure 2 right shows touch-joystick GUI for simplification of joystick control interfaces. The user can slide the smaller circle in the center in any direction within the larger circle, which has the same effect as moving the joystick in that direction. The small circle on the left side of the Jacontrol can be slid up or down to achieve the same result as twisting the joystick. The indication of up/down and clockwise/counterclockwise facilitates users to control toward the desired direction. In addition, instead of switching back and forth between finger mode and other modes during grasping tasks, the touch-joystick has two buttons for finger motions. Holding down the “Open Hand” and “Close Hand” buttons on the right side open and close the ARM’s three fingers, and the third button, “Switch to Wrist Mode”, allows the user to toggle between the translation and wrist rotation control modes. This GUI design reduces the frequency in changing modes and moreover, improves the task performance.

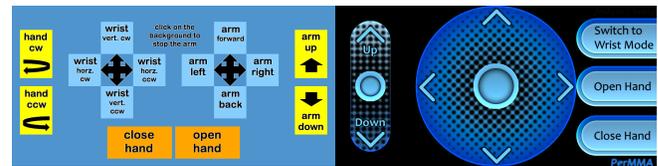


Figure 2: the GUIs: touch-keypad (left) and touch-joystick (right)

Jacontrol Program

The communication between Jacontrol App and Jacontrol program can be done through either Bluetooth or Wi-Fi. Once the communication is established, the Jacontrol program simultaneously receives control mode, finger motion, and the amount of the motion in each direction from the Jacontrol smartphone app. The program then publishes ROS topics to path planning function nodes or safety protection functions for further calculation. The computed desired ARM pose then send to the ARM and the ARM moves accordingly. If there is no other function nodes found, the message will be translated to ARM velocities and sent to the ARM directly.

Performance Evaluation

To evaluate the Jacontrol, we used the standardized ADL Task Board performance evaluation tool (C.-S. Chung, Wang, Kelleher, & Cooper, 2013). The ADL Task Board (Figure 3) is an interactive board with mounted buttons, switches, and other, similar apparatuses frequently involved in ADLs. The ADL Task Board measures the effectiveness and ease of use of different ARM interfaces: subjects use an ARM to perform different activities with the ADL Task Board (e.g., using the ARM to press a button) and record success rate and completion time for successful attempts. The ADL task board is a good method of testing ARM interfaces because it is adapted from the ISO 9241-9 (Teather, Natapov, & Jenkin, 2010), a standard, reliable

tool, and because its generalizability to any ARM interface facilitates comparisons between interfaces.



Figure 3: the Jaco ARM and ADL Task Board (left) and touch-joystick control interface and the original joystick controller (right)

Experimental Protocol

We evaluated the aforementioned two GUIs, touch-keypad and touch-joystick, in comparison with the original joystick using two tasks on the ADL task board: big yellow button and silver elevator button. The task consisted of starting the ARM at a specified point and then moving it to press the target button. One of the developers performed each task twenty times using four controllers, and all completion times were recorded. The fifteen fastest completion times for each controller were used in results calculations. A one-way repeated measures ANOVA was conducted to compare the effect of user interfaces on task completion time, ISO 9241-9 throughput and smoothness using joystick, touch-keypad, and touch-joystick.

RESULTS

The systematic reliability and communication between two devices of Jacontrol was tested for over twenty-four hours without power off. Both app and program were functional with no error and the communication has never been disconnected.

There was a statistical difference found among the user interfaces (Table 1). Post hoc tests using the Bonferroni correction revealed that completion times and throughputs are statistically different between each controller ($p < .001$ in all pairwise comparisons).

Figures 4 show the bar plot of the fifteen fastest completion times for each of the three interfaces tested when performing the big button and elevator button on the ADL Task Board. For both tasks, the order of the controllers' performance is the same—the touch-joystick had demonstrated statistically shorter completion times, followed by the joystick, and finally the touch-keypad

controller. Also notable is the touch-keypad's slow performance on the elevator button task—its mean completion time was three times slower than any of the other interfaces. In Figure 5, the touch-joystick demonstrated statistically higher ISO9241-9 throughput than other interfaces. This highest throughput suggests that the touch-joystick is the most efficient solution to the tester.

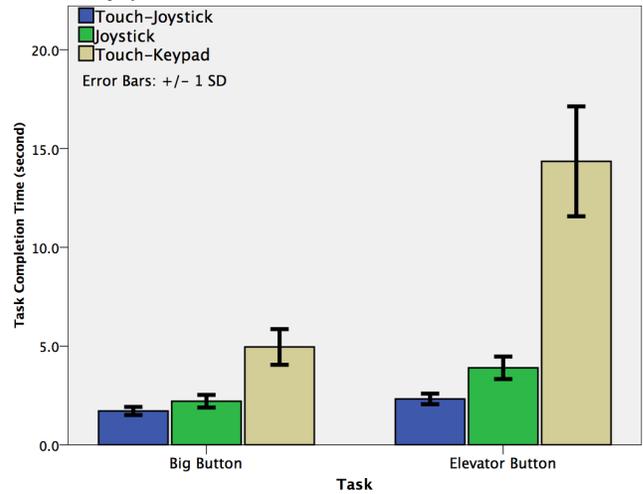


Figure 4: The error bar plot of task completion times (second) between the three control interfaces

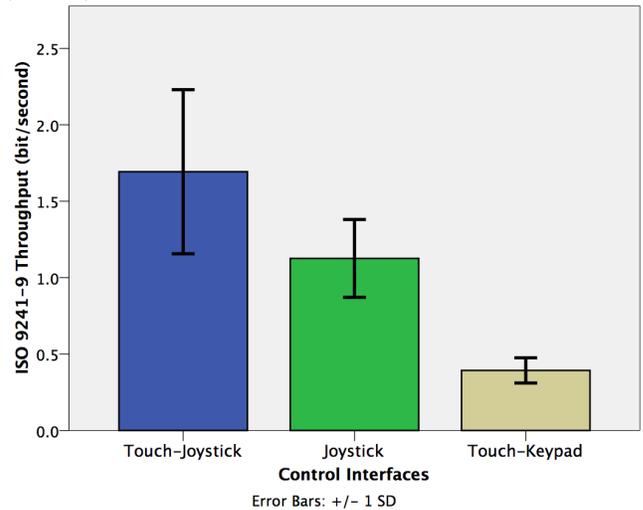


Figure 5: The comparison of ISO 9241-9 Throughput between three control interfaces.

DISCUSSION

This development is to investigate the Jacontrol framework that provides customizable smartphone GUIs and their feasibility and ARM performance in daily usage. This Jacontrol intends to overcome some challenges in current commercial physical and developing ARM control interfaces. The improvements include the customizable GUI

Table 1: Task completion time

UI (N=15)	Touch-Joystick	Touch-Keypad	Joystick	F(3,12)	p
Big Button (ID=2.034)	1.7±0.2 sec	4.9±0.9sec	2.2±0.3sec	135.337	<.001*
Elevator Button (ID=5.006)	2.3±0.3 sec	14.4±2.8sec	3.9±0.6sec	380.761	<.001*
ISO 9241-9 Throughput (bit/sec)	1.69±0.54	0.71±0.23	0.39±0.83	375.650	<.001*

that mimics commercial ARM controllers, the wire-free connection that offers remote operation and flexibility in positioning controller, the expandability to robotic autonomy and object recognition, and enhanced performance in comparison with the current joystick controller.

Over twenty-four hour reliability and stability tests suggest that the Jacontrol shows the potential in providing all day-long manipulation assistance without having troubles in the system crash or connection lost.

The touch-joystick shows statistically highest throughput and lowest task completion time among all controllers. One potential reason for the touch-joystick outperforming the joystick is the slider design. This may suggest that the slider design with indication is a more intuitive or easier motion than twisting the joystick. During the testing, we observed less up-and-down directional errors using touch-joystick. The multi-touch motions facilitate in accelerating task performance by moving three directions simultaneously.

The touch-keypad's inefficient performance should also be noted—our results suggest that this interface is not efficient at manipulating small targets even we have made improvements on the grouped functional buttons and easier accessible stop. It's design makes it easy to accelerate the ARM, but more difficult to decelerate, which our tester found unintuitive. However, this may be useful for people who cannot use the other controllers.

In conclusion, two smartphone GUIs and the Jacontrol were developed. Their ability to perform a common ADL, pushing different sized buttons, was evaluated in comparison with the default joystick interface. The results suggest the Jacontrol is a viable controller and possibly even easier to use.

One direction for future work would be conducting a clinical testing with end users and eliminating the size of the computer so it can be easily installed on the Jaco ARM.

Study Limitation

Our results are far from conclusive—more testing is needed with a larger sample size, and with a sample that contains subjects unable to use a joystick, before the Jacontrol can be proven a useful tool. However, our results suggest that the touch-joystick of the Jacontrol could be an easier to use controller for the Jaco ARM than the joystick, both for users with impaired gripping abilities and those without.

ACKNOWLEDGEMENT

This material is based upon work supported by Quality of Life Technology Engineering Research Center, National Science Foundation (Grant #0540865). The contents of this paper do not represent the views of the Department of Veterans Affairs or the United States Government.

REFERENCES

- Bach, J. R., Zeelenberg, A. P., & Winter, C. (1990). Wheelchair-mounted robot manipulators. Long term use by patients with Duchenne muscular dystrophy. *American Journal of Physical Medicine Rehabilitation Association of Academic Physiatrists*, 69(2), 55–59. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/12212433>
- Chung, C., Hannan, M. J., Wang, H., Kelleher, A. R., & Cooper, R. A. (2014). Adapted Wolf Motor Function Test for Assistive Robotic Manipulators User Interfaces : A Pilot Study. In *RESNA Annual Conference*.
- Chung, C.-S., Wang, H., & Cooper, R. A. (2013a). Autonomous function of wheelchair-mounted robotic manipulators to perform daily activities. *IEEE ... International Conference on Rehabilitation Robotics : [proceedings]*, 2013, 1–6. doi:10.1109/ICORR.2013.6650378
- Chung, C.-S., Wang, H., & Cooper, R. A. (2013b). Functional assessment and performance evaluation for assistive robotic manipulators: Literature review. *The Journal of Spinal Cord Medicine*, 36(4), 273–89. doi:10.1179/2045772313Y.0000000132
- Chung, C.-S., Wang, H., Kelleher, A., & Cooper, R. A. (2013). Development of a Standardized Performance Evaluation ADL Task Board for Assistive Robotic Manipulators. In *Proceedings of the Rehabilitation Engineering and Assistive Technology Society of North America Conference*. Seattle, WA.
- Dicianno, B. E., Parmanto, B., Fairman, A. D., Crytzer, T. M., Yu, D. X., Pramana, G., ... Petrazzi, A. a. (2014). Perspectives on the Evolution of Mobile (mHealth) Technologies and Application to Rehabilitation. *Physical Therapy*. doi:10.2522/ptj.20130534
- Jiang, H., Zhang, T., Wachs, J. P., & Duerstock, B. S. (2014). Autonomous Performance of Multistep Activities with a Wheelchair Mounted Robotic Manipulator Using Body Dependent Positioning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop on Assistive Robotics for Individuals with Disabilities: HRI Issues and Beyond*. Chicago, Illinois, USA. Retrieved from <http://www.haewonpark.com/IROS2014-ARHRI/files/IROS2014-WS-ARHRI-S15.pdf>
- Teather, R. J., Natapov, D., & Jenkin, M. (2010). Evaluating haptic feedback in virtual environments using ISO 9241–9. In *2010 IEEE Virtual Reality Conference (VR)* (pp. 307–308). IEEE. doi:10.1109/VR.2010.5444753
- Tsui, K. M., Kim, D.-J., Behal, A., Kontak, D., & Yanco, H. A. (2011). I Want That ”: Human-in-the-Loop Control of a Wheelchair-Mounted Robotic Arm. *Applied Bionics and Biomechanics*, 8(1), 127–147. doi:10.3233/ABB-2011-0004
- Wang, H., Xu, J., Grindle, G., Vazquez, J., Salatin, B., Kelleher, A., ... Cooper, R. a. (2013). Performance evaluation of the Personal Mobility and Manipulation Appliance (PerMMA). *Medical Engineering & Physics*, 35(11), 1613–9. doi:10.1016/j.medengphy.2013.05.008
- Wu, Y., Liu, H., Brown, J., Kelleher, A., & Cooper, R. A. (2013). A Smartphone Application for Improving Powered Seat Functions Usage : A Preliminary Test. In *RESNA Annual Conference* (pp. 2–5).