Development of Smartphone-Based Human-Robot Interfaces for Individuals with Disabilities

Lei Wu, Redwan Alqasemi, and Rajiv Dubey

***Abstract*— Persons with disabilities often rely on caregivers or family members to assist in their daily living activities. Robotic assistants can provide an alternative solution if intuitive user interfaces are designed for simple operations. Current human-robot interfaces are still far from being able to operate in an intuitive way when used for complex activities of daily living (ADL). In this era of smartphones that are packed with sensors, such as accelerometers, gyroscopes and a precise touch screen, robot controls can be interfaced with smartphones to capture the user's intended operation of the robot assistant. In this paper, we review the current popular human-robot interfaces, and we present three novel human-robot smartphone-based interfaces to operate a robotic arm for assisting persons with disabilities in their ADL tasks. Useful smartphone data, including 3 dimensional orientation and 2 dimensional touchscreen positions, are used as control variables to the robot motion in Cartesian teleoperation. In this paper, we present the three control interfaces, their implementation on a smartphone to control a robotic arm, and a comparison between the results on using the three interfaces for three different ADL tasks. The developed interfaces provide intuitiveness, low cost, and environmental adaptability.**

I. INTRODUCTION

There are more than 28 million people in the US who have some form of physical disability and require assistance [1]. Providing robotic assistants with easy and intuitive user interfaces to assist with activities of daily living (ADL) can improve their quality of life and lift some of the burden on caregivers and family members. Several research groups have worked on human-robot interfaces to enhance the usability of service robots, and the most popular interfaces are reviewed and compared in table I. From the reviews, we can see that there are different shortcomings in each method that make it more difficult or impossible to complete complex ADL tasks. In this paper, our focus is to build novel interfaces based on smartphones, which are readily available devices, eliminating as many shortcomings as possible in current existing human-robot interfaces to provide a simple and easy to use interface for users to control the robotic arm and perform their daily living activities. Our novel interfaces should achieve the following objectives: 1. intuitive interaction with the least amount of buttons such that minimum or no training is needed; 2. precise and responsive enough so that healthy people and persons with disabilities can complete complex ADL tasks in an unstructured environment with minimum or no frustration; 3. no calibration is needed; 4. low cost; 5. lightweight and wireless. These goals will also distinguish our work from existing methods.

II. CONTROL INTERFACES AND THEORIES

In this work, three different control interfaces have been developed for controlling a robotic arm using a smartphone. One uses a single touch anywhere on the screen (**One Button**), and the other two use three buttons on the screen (**Three Buttons** and **Tilt**). For user safety, all control interfaces require the users to keep their finger on the phone screen while they control the robot, and will stop moving the arm if the user releases their finger from the screen. When

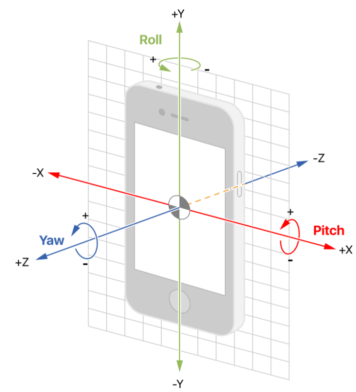


Fig. 1: Smartphone coordinate system

operating the robotic arm in the 3D Cartesian space, the user needs to provide the robot controller with six user inputs, three translation velocity values and three rotation velocity values, to control all Cartesian degrees of freedom (DoF). A typical smartphone with accelerometers and gyroscope sensors can supply 3 degrees of freedom of its own pose, which are directly related to the *pitch*, *roll* and *yaw* of the phone. These three values can be used as user input values to provide three of the six DoF values required to control the robot end-effector in the Cartesian coordinates. Furthermore, when the user touches the screen, the position of the touch on the screen can provide x and y directions on the screen surface, which can be used to control two more of the six DoF values required for Cartesian space motion. This provides a total of five input values out of the six needed input values from the user as shown in Figure 1. We created a smartphone application that has several user interfaces. The application will save the initial touch position and orientation when the user places their finger on the screen, then as the user slides that finger on the screen or tilts the smartphone, the application is continuously using the current reading minus the initial reading, and the result is transmitted to the PC over WIFI through TCP/IP protocol at

TABLE I: Comparison of the state of the art human-robot interfaces

Human-Robot Interface Method	**User's Interaction**	**Advantages**	**Disadvantages**
EMG based [2]–[5]	User wears EMG device on their lower arm	Amputees are able to use it	User needs to be trained, needs calibration, cannot control translation and orientation at the same time, high cost.
Kinect based [6]–[8]	Computer will process Kinect 3D images, and provide many predefined task options, such as grasping, get near and drop.	More autonomous than other methods, good for users who have difficulties with a touch screen, low cost, commercially available.	Not precise, not responsive, cannot see transparent objects, needs careful calibration, camera view can be obstructed by the robot arm, can only recognize predefined objects, limited task options.
Leap motion based [9]–[11]	Move hands above the device. Device can track the hand position within a specific range.	Low cost, commercial available, can recognize both hands using a single device.	Accuracy of the hand pose requires the user's hand to be constantly suspended at a specific distance from the device, which may cause fatigue and abandonment.
IMU based [12]–[14]	User wears an IMU device on their hands or head. The robotic arm is controlled by tilting the device.	Precise	Not commercial available, needs separate battery pack, can only control 3 DoF at a time, needs to switch modes to control all 6 DoF, not very intuitive for controlling XYZ translation.
Brain computer interfaces [15]–[17]	User wears a BCI input device, and reacts to a stimulation patterns displayed on a screen.	Works good for users whose motor functions are completely paralyzed, and their cognitive abilities are intact.	Needs lengthy calibration, needs training, needs focused attention, causes fatigue, high cost, low responsiveness.
Joysticks based [18], [19]	One up button one down button, one control stick for X and Y, one stick for pitch and roll, and one stick for yaw	Low cost, commercially available, very precise and responsive.	Complicated 6 DoF control, not intuitive, requires both hands working together, which can be difficult for a person with upper limb disability.
Touch screen/smartphone based [20]–[22]	Touch buttons to open/close gripper, drag virtual stick to control X/Y, drag slider to control Z.	Commercially available, low cost, wireless, light weight.	Too many buttons and sliders on the screen, cannot control orientation of the gripper. Can be useful for very simple tasks.
Voice control [23]	Computer will recognize voice commands, including up, down, left, right, open and close	No need to use hands.	Not responsive, cannot control orientation of the gripper, very difficult to achieve ADLs.
Physical human-robot interaction [24], [25]	Users use their hand to hold a master robotic arm end-effector. The system will mirror the motion to the slave robot arm.	Responsive, precise, intuitive, can control 6 DoF at the same time.	High cost, heavy device, difficult to carry.
Haptic device [26]	Use Phantom Omni Premium to read the position and orientation of human hand, then directly mapping the motion to the robot hand.	Precise, responsive, can control 6 DoF at the same time.	High cost, heavy device with a lot of wires, difficult to carry.

around 90Hz. If the user releases the finger from the screen, it will send a stop signal to the robot computer. If the user starts touching the screen again, the initial position and orientation will be updated. So the output signal of the smartphone is not velocities, it is rather displacement of positions and angles. This control pattern is similar to how the joystick controls a wheelchair, the more you push the joystick the higher the wheelchair speed is. Equation 1 shows how the phone output command is calculated.

$$\begin{bmatrix} \Delta x_{phone} \\ \Delta y_{phone} \\ \Delta pitch_{phone} \\ \Delta roll_{phone} \\ \Delta yaw_{phone} \end{bmatrix} = \begin{bmatrix} x \\ y \\ pitch \\ roll \\ yaw \end{bmatrix}_{current} - \begin{bmatrix} x \\ y \\ pitch \\ roll \\ yaw \end{bmatrix}_{initial} \tag{1}$$

where *current* is the current position and orientation value of the phone, and *initial* is the position and orientation data recorded when the user starts touching the screen. Using relative phone pose reading will give users the freedom of using it in any gesture, such as lying on the bed or sitting on the sofa.

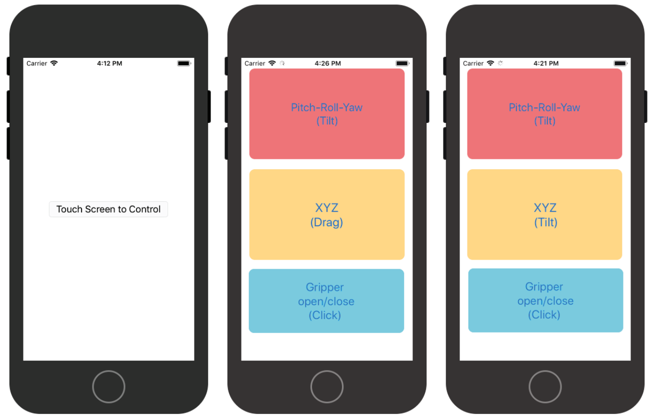
In order to utilize these phone output commands in controlling end-effector velocities, we use a diagonal square matrix C that includes coefficients that represent motion sensitivity gains for each motion direction, and unit converters from position and orientation to linear and angular velocities. Equation 2 shows the diagonal coefficient matrix C.

$$C = \begin{bmatrix} c_x & 0 & 0 & 0 & 0 & 0 \\ 0 & c_y & 0 & 0 & 0 & 0 \\ 0 & 0 & c_z & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{\omega_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{\omega_x} & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{\omega_z} \end{bmatrix} \tag{2}$$

where c_x, c_y, c_z, c_{ω_x}, c_{ω_y}, c_{ω_z} are scaling coefficients that are used as gains for the smartphone control vector, and unit conversion factors that convert displacement to speed.

A. One Button Interface

We decided to design an intuitive interface that is different from any other existing touchscreen/smartphone based solutions which have too many buttons. In this control interface, there is only one button, which is placed throughout the whole screen, that is required to control all 6 DoF of the end-effector. The user may start touching anywhere on the screen, and then drag their finger on screen to control the X and Y translations of the robot end-effector. Additionally, the user can control the robot end-effector orientation simultaneously



(a) **One Button**: slide translation & tilt rotation(combine) (b) **Three Buttons**: slide translation & tilt rotation (separate) (c) **Tilt**: tilt translation & tilt rotation (separate)

Fig. 2: Developed three interfaces

by tilting the phone in a similar fashion as they want the end-effector to be oriented in the Cartesian space. Figure 2a shows the phone application view of the "**One Button**" interface. Equation 3 represents the mapping between the smartphone application output vector and the velocity vector of the robot end-effector. For the translation part, the farther you slide your finger on the screen, the faster the end-effector's linear velocity is. For the rotational part, the more you tilt the smartphone, the faster the end-effector's angular velocity is.

$$v_e = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}_e = C \cdot \begin{bmatrix} \Delta y_{phone} \\ \Delta x_{phone} \\ 0 \\ \Delta roll_{phone} \\ \Delta pitch_{phone} \\ \Delta yaw_{phone} \end{bmatrix} \quad (3)$$

where v_e is the robot end-effector velocity vector, $\dot{x}$, $\dot{y}$, $\dot{z}$ are translational speed in mm/s. ω_x, ω_y, ω_z are rotational speed in rad/s. Δx_{phone}, Δy_{phone} are the planar distances between the initial touch position to the current touch position on the screen. $\Delta pitch_{phone}$, $\Delta roll_{phone}$ and Δyaw_{phone} are the 3D orientation angles' differences between the phone pose at the initial touch and the phone pose at the current touch on the screen. The unit of Δx_{phone}, Δy_{phone} is pixels, and the unit of $\Delta pitch_{phone}$, $\Delta roll_{phone}$ and Δyaw_{phone} is degrees. In this application, c_x=0.25$mm/pixels*s$, c_y=0.3$mm/pixels*s$, c_{ω_x}=0.003$rad/degrees*s$, c_{ω_y}=c_{ω_z}=0.0025$rad/degrees*s$. These gain values were found by assigning a rough estimation first, then adjusting it according to users' preference. The reason why the gain for sliding in the x direction is larger than that for sliding in the y direction is because the width of a smartphone is typically smaller than the length, and choosing a different gain value will yield a better range of velocities for the end-effector.

Notice that the motion in the Z direction is set to zero as a default since there are only 5 user input values from the phone as shown in equation 1. In order to control the robot in the Cartesian Z direction without adding more buttons and sacrificing **One Button** design intention, we created the Z control mode. If the user would like to move the robot end-effector in the Z direction, the user needs to pitch the phone up prior to touching the screen so that the Z control mode is activated. If Z control mode is activated, only Y(left and right) and Z(up and down) axes of the robot end-effector can be controlled using the finger's planar drag on the touch screen as shown in equation 4.

$$v_e = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}_e = C \cdot \begin{bmatrix} 0 \\ \Delta x_{phone} \\ \Delta y_{phone} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4)$$

In this case, c_y=0.3$mm/pixels*s$, c_z=0.25$mm/pixels*s$.

The user can quit Z control mode by releasing their finger from the touch screen, and pitching the phone down to the floor plane. This alternating mapping of the finger dragging on the touch screen can provide intuitive control of the 3D position of the end-effector using the planar touch screen. To toggle the status of gripper between the "open" and "close" positions, double-touching anywhere on the touch screen will alternate between the two gripper positions.

B. Three Buttons Interface

Some people prefer maneuvering translational motion and rotational motion independently, so **Three Buttons** interface is designed. It uses three buttons on the screen instead of a single button. Each one of these buttons, when touched, controls a specific Cartesian motion of the gripper. The first button only activates the Cartesian orientation control of the robot end-effector. The second button only activates the Cartesian position control of the end-effector. The third button toggles the "open" and "close" positions of the end-effector. Figure 2b shows the user interface of the developed application. As shown, when Pitch-Roll-Yaw button is touched, tilting the phone in any direction will provide three DoF Cartesian orientation values for the robot end-effector to move to a similar orientation. In this case, the position of the end-effector will not be changed, dragging the finger on the touch screen will not cause any motion. Equation 5 shows the Cartesian velocity of the robot end-effector when Pitch-Roll-Yaw button is touched.

$$v_e = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}_e = C \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ \Delta roll_{phone} \\ \Delta pitch_{phone} \\ \Delta yaw_{phone} \end{bmatrix} \quad (5)$$

In this case, c_{ω_x} = 0.003 $rad/degrees*s$, c_{ω_y} = c_{ω_z} = 0.0025 $rad/degrees*s$.

When XYZ button is touched, the three DoF Cartesian position values of the robot end-effector are provided through planar finger dragging on the touch screen for the X and Y values, and tilting the phone up and down (pitch direction) for the Z value. In this case, the orientation of the robot end-effector will not be changed, as the phone's roll and yaw motions will not cause any motion. Equation 6 shows the Cartesian velocity of the robot end-effector when XYZ button is touched.

$$v_e = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}_e = C \cdot \begin{bmatrix} \Delta y_{phone} \\ \Delta x_{phone} \\ \Delta pitch_{phone} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (6)$$

In this case, c_x=0.25$mm/pixels * s$, c_y=0.3$mm/pixels * s$, c_z=0.7$mm/degrees * s$.

When the Gripper button is touched, the user will be able to control the "open" and "close" positions of the robot end-effector. A single touch will toggle the status of the gripper.

C. Tilt Interface

Some persons with upper extremity disabilities may have the ability to move their hand, but have limited finger motion and dexterity that makes it difficult to move their thumb across a smartphone screen. For such cases, we developed the third control interface, **Tilt**. This interface will eliminate the need for finger motion on the screen and will use only tilting to control all 6 DoF of the robotic arm. As shown in Figure 2c, this **Tilt** control interface also shows 3 buttons on the screen, similar to the **Three Button** interface. The function and control logic of the first button and third button are the same as that of the **Three Buttons** interface. The first button (Pitch-Roll-Yaw Tilt) uses the same mapping for the Cartesian velocity of the robot end-effector as in Equation 5. However, the second button(XYZ Tilt) uses different control logic. Equation 7 shows the Cartesian velocity vector of the robot end-effector when XYZ button is touched.

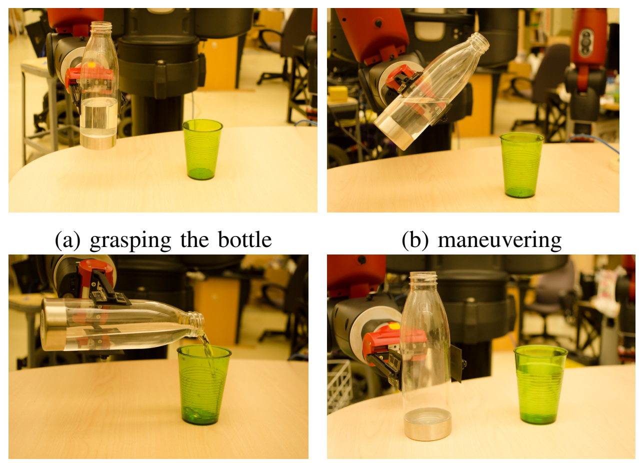
$$v_e = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}_e = C \cdot \begin{bmatrix} \Delta roll_{phone} \\ \Delta yaw_{phone} \\ \Delta pitch_{phone} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (7)$$

In this case, c_x=c_y=c_z=0.7$mm/degrees * s$. Inspired by how a screwdriver drives a screw, we use roll rotation of the phone to control the forward and backward motions of the end-effector. Rotating clockwise refers to moving forward, and rotating counterclockwise refers to moving backwards.

III. Experimental setup

To perform preliminary testing of our new interfaces with ADL tasks, we recruited subjects that are not familiar with the system and have no prior experience in controlling robotic arms for ADL tasks. It is important to mention here that these tests are not meant to provide statistical significance, it is rather meant to provide feedback for adjustment of gains and to collect preliminary data on the metrics mentioned later in this paper. For that purpose, we recruited a small number of human subjects to perform three specific ADL tasks. These subjects are three healthy subjects, and one subject with spinal cord injury who has lower body and upper limbs disabilities. Future work will include recruiting a significant number of subjects and performing clinical testing that can provide statistically significant data, and we will perform power analysis that we will publish in a future publication. The study was approved by the Internal Review Board under IRB#Pro00040871. All of the subjects have no experience in using our interfaces.

Three ADL tasks were selected for this test, based on a prior survey we conducted for persons with physical disabilities to find the most common ADL tasks for which they can use robotic assistance to perform. The first task is a water pouring task, in which, users need to control the robot arm to grasp a bottle of water and maneuver the bottle to pour water into a cup without spilling. As shown in Figure 3, the user controls the robotic arm using our developed interfaces to grasp the water bottle and carefully pour water into a cup, then place the water bottle back on top of the desk.



(a) grasping the bottle (b) maneuvering

(c) pouring water into the bottle (d) placing the bottle on the table

Fig. 3: First ADL task: Water pouring

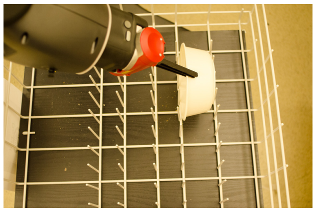
The second task is a plate&bowl pick-and-place task. As shown in Figure 4, the user controls the robotic arm to grasp a plate from the dishwasher and place it on top of the desk, then go back to the dish washer, grasp a bowl and place it on top of the plate.

The third task is pepper&salt grasping task for food seasoning, which includes opening cabinet doors and adding contents to a dining plate. As shown in Figure 5, the user controls the robotic arm to open the door of a cabinet, grasp a can of pepper, shake it onto the dining plate, and place the can back into the cabinet. Then grasp a can of salt, shake it onto the dining plate, and place the can back into the cabinet. Finally, the user closes the door of the cabinet.

The hardware used for the experimental evaluation are the Baxter robot (for it's low cost as an affordable assistant robot), a PC (which is a standard unit available in most homes and workplaces), and an iPhone SE (which is a typical

(a) grasping the plate

(b) placing the plate on the tabletop



(c) grasping the bowl

(d) placing the bowl on the plate

Fig. 4: Second ADL task: Pick and place plate&bowl from the dish washer to the tabletop

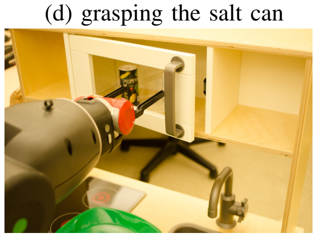
(a) opening the cabinet door

(b) grasping the pepper can

(c) shaking pepper onto the plate

(d) grasping the salt can

(e) shaking salt onto the plate



(f) closing the cabinet door

Fig. 5: Third ADL task: Seasoning the food with Pepper and Salt

cost effective smartphone). Our interfaces can be applied on most robotic arms and smartphones. The smartphone sends the user input values to the PC through WiFi TCP/IP communication. Kinematics and redundancy resolution methods to control the end-effector in the Cartesian space are discussed in a separate publication [27], which is not the focus of this paper.

IV. Evaluation and Results

We collected both quantitative and qualitative data from human subjects' tests with the three ADL tasks. For the quantitative data, we recorded the time required for the users to perform each task, the lower performance time means the task was easier to complete using that interface. We also measured the total distance traveled by the robot end-effector, the lower distance traveled means the lower unnecessary motion executed when performing the task using that interface. All three interfaces were tested in reference to the ground(**G**) coordinate frame (**One Button_G**,**Three Buttons_G**, **Tilt_G**) and in reference to the end-effector (**E**) coordinate frame (**One Button_E**,**Three Buttons_E**, **Tilt_E**). Each user performed each task three times for each user interface and for each control reference frame. Since we have 3 user interfaces, 2 control reference frames, and 3 ADL tasks, the total number of experiments performed by each user was 54. After completing each ADL task, we asked the user to answer one qualitative question, which is: "How intuitive the interface was to complete this task?". Users answered the question on a scale from 0-10, where 10 is "very intuitive", and 0 is "very difficult to use".

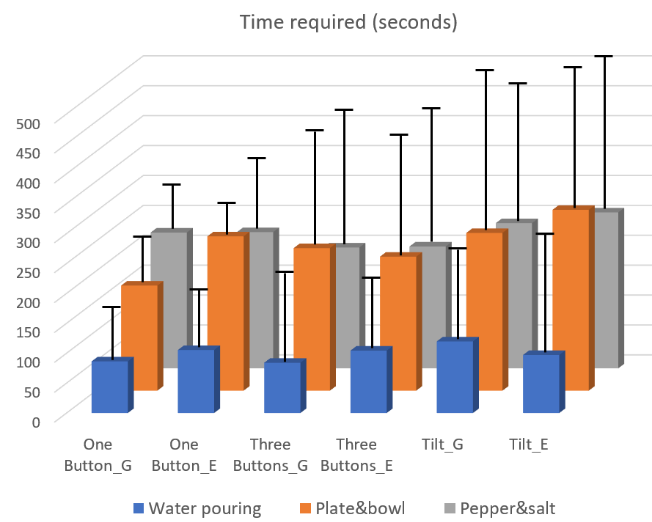


Fig. 6: Time required to complete each task (the lower the better). Subscripts _G and _E refer to the control in Ground reference and End-effector reference, respectively.

Figure 6 compares the time required to complete each of the three ADL tasks using each interface. The solid columns represent averages of all recorded time segments from all the healthy subjects in each interface and task, and the error bars represent the data from the subject with disabilities. In the water pouring task, **One Button** and **Three Buttons** control in the ground reference frame required the least amount of time, while **Tilt** control in the ground reference frame required the highest amount of time. In the plate&bowl pick-and-place task, **One Button** control in the ground reference frame required the least amount of time, while **Tilt** control in the end-effector reference frame required the highest amount of time. In the pepper&salt grasping task, **Three Buttons** control in both the ground and the end-effector reference frames achieved the shortest time, while **Tilt** control in the end-effector reference frame required the highest amount of time. Compared to the joystick based solution presented in [19], our methods required 25% to 35% of the time. The main reason for this significant time savings is that they used a joystick, which controls 2 DoF at a time, and they needed to switch control modes frequently, which required much more time and effort to perform the task. However, our one button interface for controlling all 6 DoF has eliminated all unnecessary mode switching time.

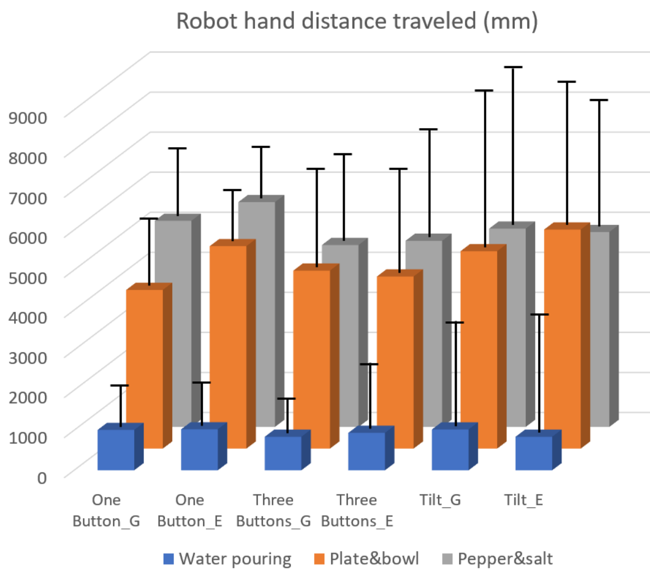


Fig. 7: Robot end-effector distance traveled (the lower the better). Subscripts _G and _E refer to the control in Ground reference and End-effector reference, respectively.

Figure 7 compares the distance traveled to complete each ADL task using each user interface, the lower distance traveled means that users had more precise motion in using that interface and can easily eliminate unnecessary motions. The solid columns represent data averages of all the healthy subjects in each interface and task, and the error bars represent the data from the subject with disabilities. In water pouring task, **Three Buttons** control in the ground reference frame achieved the least travel distance among the other interfaces, while **One Button** control in both the ground and the end-effector reference frames achieved the highest traveled distance among the other interfaces. In the plate&bowl pick and place task, **One Button** control in the ground reference frame achieved the least travel distance among the other interfaces due to the fact that the plate&bowl pick and place task requires a lot of rotational maneuvering, while**Tilt** control in the end-effector reference frame achieved the highest traveled distance among the other interfaces. **One Button** is the best interface choice for this situation, since it can control all 6 DoF at the same time. In the pepper&salt grasping task, **Three Buttons** control in both the ground and the end-effector reference frames required the least travel distance, while **One Button** control in the end-effector reference frame traveled the highest distance among the other interfaces.

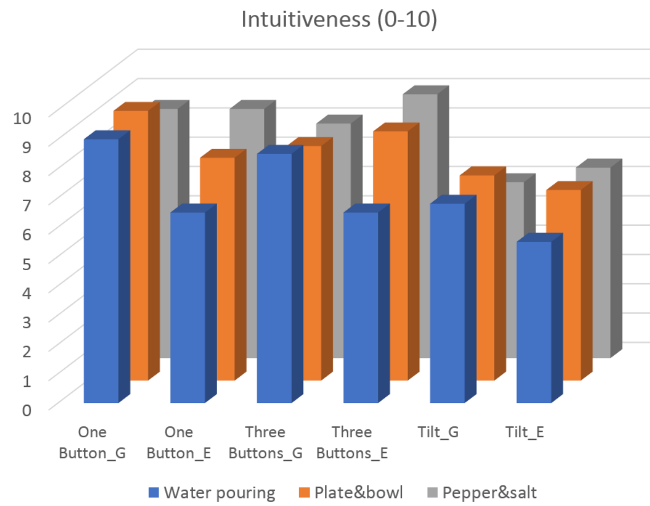


Fig. 8: How intuitive the interface was in each task (The higher the better). Subscripts _G and _E refer to the control in Ground reference and End-effector reference, respectively.

Figure 8 compares the users' feedback of average intuitiveness for each interface in the three ADL tasks. In the water pouring task, both **One Button** and **Three Buttons** control in the ground reference frame have high intuitiveness ratings, while **Tilt** control in the end-effector reference frame has the least intuitiveness rating. In the plate&bowl pick and place task, users rated **One Button** control in the ground reference frame the highest intuitiveness rating, and **Tilt** control in the end-effector reference frame the lowest intuitiveness rating. In the pepper&salt grasping task, both **One Button** and **Three Buttons** control in the end-effector reference frame have high intuitiveness ratings, while **Tilt** control in the ground reference frame has the low intuitiveness rating. Since this task required the user to open the door of the cabinet and grasp the pepper and salt cans, control in the end-effector reference frame is easier and more straightforward than the control in the ground reference frame.

V. CONCLUSIONS

Our proposed novel smartphone based human-robot control interfaces achieved intuitive and effortless control of the robotic arm to fulfill relatively complex activities of daily living tasks that other methods can fail or be difficult to do. All users, without any pre-experiences, were able to get used to our intuitive smartphone based interfaces very quickly and were able to successfully complete the challenging ADL tasks without any training. Compared to most widely used joystick based interfaces, our interfaces required only one hand, it was very intuitive especially when controlling the rotational motion of the robot arm. Compared to all other existing interfaces mentioned in table I, many good advantages were achieved in this research: (1) It does not need calibration, (2) It does not need training, users can get used to these interfaces in few minutes, (3) Very low cost,

(4) Responsive and precise, (5) Effortless, it is wireless, light weight and uses relative phone pose reading, so that users can use this interface in any gesture, such as sitting on the wheelchair or lying on the bed, (6) Safe, the robotic arm will stop moving immediately once the finger released from the phone screen.

VI. Future Work

From users' feedback, there are many areas that can be improved. First, for maneuvering the gripper in forward/backward translational motion, and in pitch or roll rotational motions, users preferred to maneuver in the end-effector reference frame. For maneuvering the gripper in left/right, up/down translational motions and in yaw rotational motion, users preferred to maneuver in the ground reference frame. We will develop a novel hybrid control coordinate system and its corresponding kinematics equations of the robotic arm so that we can achieve hybrid reference frame control for improved intuitiveness and more natural motion mapping. Second, users preferred **One Button** interface because it has only one button, and it eliminated the need for visual feedback. Users suggested adding haptic feedback when touching the buttons in **Three Buttons** and **Tilt** interfaces in order to eliminate the need for looking at the screen when touching the buttons. Additionally, users suggested adding audio feedback to all three interfaces when toggling the gripper status to open and close. We plan to add these features in the future. Furthermore, we plan to perform a clinical study and recruit a significant number of healthy subjects and subjects with disabilities to perform more complex ADL tasks and collect statistically significant data on our interfaces, including a new novel hybrid control coordinate system that is under development.

References

[1] E. Lauer and A. Houtenville, "2017 annual disability statistics supplement." *Institute on Disability, University of New Hampshire*, 2018.

[2] C. Yang, C. Zeng, P. Liang, Z. Li, R. Li, and C.-Y. Su, "Interface design of a physical human–robot interaction system for human impedance adaptive skill transfer," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 329–340, 2017.

[3] R. Meattini, S. Benatti, U. Scarcia, D. De Gregorio, L. Benini, and C. Melchiorri, "An semg-based human–robot interface for robotic hands using machine learning and synergies," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 8, no. 7, pp. 1149–1158, 2018.

[4] U. C. Allard, C. L. Fall, A. Campeau-Lecours, C. Gosselin, F. Laviolette, and B. Gosselin, "Transfer learning for semg hand gestures recognition using convolutional neural networks," *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1663–1668, 2017.

[5] C. L. Fall, G. Gagnon-Turcotte, J.-F. Dube, J. S. Gagne, Y. Delisle, A. Campeau-Lecours, C. Gosselin, and B. Gosselin, "Wireless semg-based body–machine interface for assistive technology devices," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, pp. 967–977, 2017.

[6] C. Bousquet-Jette, S. Achiche, D. Beaini, Y. S. L.-K. Cio, C. Leblond-Ménard, and M. Raison, "Fast scene analysis using vision and artificial intelligence for object prehension by an assistive robot," *Eng. Appl. of AI*, vol. 63, pp. 33–44, 2017.

[7] H. W. Ka, D. Ding, and R. A. Cooper, "Three dimensional computer vision-based alternative control method for assistive robotic manipulator," in *ICRA 2016*, 2016.

[8] D. J. Butler, S. Elliott, and M. Cakmak, "Interactive scene segmentation for efficient human-in-the-loop robot manipulation," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2572–2579, 2017.

[9] D. Bassily, C. Georgoulas, J. Güttler, T. Linner, and T. Bock, "Intuitive and adaptive robotic arm manipulation using the leap motion controller," in *ISR 2014*, 2014.

[10] G. Du and P. Zhang, "A markerless human–robot interface using particle filter and kalman filter for dual robots," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2257–2264, 2015.

[11] Y. Lin, S. Song, and M. Q.-H. Meng, "The implementation of augmented reality in a robotic teleoperation system," *2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 134–139, 2016.

[12] T. L. Baldi, G. Spagnoletti, M. Dragusanu, and D. Prattichizzo, "Design of a wearable interface for lightweight robotic arm for people with mobility impairments," *2017 International Conference on Rehabilitation Robotics (ICORR)*, pp. 1567–1573, 2017.

[13] S. Jain, A. Farshchiansadegh, A. Broad, F. Abdollahi, F. A. Mussa-Ivaldi, and B. Argall, "Assistive robotic manipulation through shared autonomy and a body-machine interface," *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*, pp. 526–531, 2015.

[14] C. L. Fall, P. Turgeon, A. Campeau-Lecours, V. Maheu, M. Boukadoum, S. Roy, D. Massicotte, C. Gosselin, and B. Gosselin, "Intuitive wireless control of a robotic arm for people living with an upper body disability," *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4399–4402, 2015.

[15] K. Muelling, A. Venkatraman, J.-S. Valois, J. Downey, J. Weiss, S. Javdani, M. Hebert, A. B. Schwartz, J. L. Collinger, and J. A. Bagnell, "Autonomy infused teleoperation with application to brain computer interface controlled manipulation," *Autonomous Robots*, vol. 41, pp. 1401–1422, 2017.

[16] A. Mohammed and L. Wang, "Brainwaves driven human-robot collaborative assembly," *CIRP annals*, vol. 67, no. 1, pp. 13–16, 2018.

[17] C.-Y. Chiu, A. K. Singh, Y.-K. Wang, J.-T. King, and C.-J. Lin, "A wireless steady state visually evoked potential-based bci eating assistive system," *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 3003–3007, 2017.

[18] A. Campeau-Lecours, U. Côté-Allard, D.-S. Vu, F. Routhier, B. Gosselin, and C. Gosselin, "Intuitive adaptive orientation control for enhanced human–robot interaction," *IEEE Transactions on Robotics*, vol. 35, pp. 509–520, 2019.

[19] L. V. Herlant, R. M. Holladay, and S. S. Srinivasa, "Assistive teleoperation of robot arms via automatic time-optimal mode switching," in *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*. IEEE Press, 2016, pp. 35–42.

[20] C.-S. Chung, H. W. Ka, H. Wang, D. Ding, A. Kelleher, and R. A. Cooper, "Performance evaluation of a mobile touchscreen interface for assistive robotic manipulators: A pilot study." *Topics in spinal cord injury rehabilitation*, vol. 23 2, pp. 131–139, 2017.

[21] C. Parga, X. Li, and W. Yu, "Tele-manipulation of robot arm with smartphone," in *2013 6th International Symposium on Resilient Control Systems (ISRCS)*. IEEE, 2013, pp. 60–65.

[22] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, S. Savarese, and L. Fei-Fei, "Roboturk: A crowdsourcing platform for robotic skill learning through imitation," in *CoRL*, 2018.

[23] S. J. Poirier, F. Routhier, and A. Campeau-Lecours, "Voice control interface prototype for assistive robots for people living with upper limb disabilities," *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, pp. 46–52, 2019.

[24] S. Haddadin and E. A. Croft, "Physical human-robot interaction," in *Springer Handbook of Robotics, 2nd Ed.*, 2016.

[25] E. Mariotti, E. Magrini, and A. D. Luca, "Admittance control for human-robot interaction using an industrial robot equipped with a f/t sensor," *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6130–6136, 2019.

[26] M. H. Vu and U. J. Na, "A new 6-dof haptic device for teleoperation of 6-dof serial robots," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, pp. 3510–3523, 2011.

[27] R. M. Alqasemi, "Maximizing manipulation capabilities of persons with disabilities using a smart 9-degree-of-freedom wheelchair-mounted robotic arm system," 2007.