2014 IEEE/RSJ International Conference on Intelligent Robots and Systems
Workshop on Assistive Robotics for Individuals with Disabilities: HRI Issues and Beyond
September 14, 2014, Chicago, Illinois, USA

A Single-Switch Scanning Interface for Robot Control by Quadriplegics.

Varvara Kalokyri[1], Rahul Shome[1], Isaac Yochelson[1] and Kostas E. Bekris [1]

***Abstract*— Robotic technology has promised considerable help to people with severe motor and communicative disabilities. However, due to motor and speech impairments these people are not able to handle commercial products with common interfaces for moving a particular robotic device. Scanning interfaces are used extensively for people with severe impairments and to this end, we present in this paper, a single-switch web-based scanning application for tablets for controlling a robotic arm only by controlling the 3D position of its end-effector. We present a comparison of our user interface with an already, proposed similar solution [4] which controls the robotic arm by controlling all the joints of the robotic arm independently. As a preliminary study, we developed both of these user interfaces and we conducted experiments with fully able subjects so as to evaluate which human-robot interface is better in terms of supporting the easiest and the most straightforward communication between robots and a motor-impaired individual.**

I. INTRODUCTION

People with speech and motor disorders face problems in expressing themselves in an easy and intelligible way. The disabilities may stem from a variety of medical conditions, such as muscular dystrophy, Parkinson's disease, cerebral palsy, spinal cord injuries as well as neuromuscular diseases, affecting hundreds of thousands of people around the world.

Service robots are an important stepping stone in helping these people improving their quality of life. With the manipulation of a robot, people with impairments are able to execute a series of everyday tasks, which otherwise could not be performed without the assistance of an able-bodied individual. However, without the appropriate user interface specially designed for these people, the use of a service robot would be impossible.

In this work, our target group is people with severe motor and speech impairments who can only interact using a single-switch interface. The user interfaces proposed for this group of people in the relevant literature are scanning interfaces [1]. These people can manage such interfaces in order to move the robotic arms of a service robot by using specific input devices, the access switches. Access switches are specially designed devices that require reduced motor control in order to operate. Any active body part of the user including hand, foot, mouth, or head can be used to operate such switches.

Our purpose in this work is to develop two different web-based single-switch scanning interfaces and find which is the most intuitive and easy to communicate interface for robot control by quadriplegics. The reason why we built web-based scanning interfaces is due to the user-friendly, platform agnostic and universally accessible and interactive environment, that is offered by web-enabled robotics solutions. Users of web-based robotic interfaces are able to access robots remotely through the internet and control robots through an interface on a personal computer or even better on a tablet while they are inside their house or even outside of it.

The two human-robot interfaces that we developed are: an interface with which the user is able to control only the 3D position of the end-effector of the robotic arm and an interface with which the user is able to control all the different joints of the arm indepedently. In comparison with other implemented solutions which propose the direct manipulation of all the joints of the robotic arm independently, we will show by conducting experiments, that the interface which controls the 3D position of the end effector is better than the other solutions proposed in the literature. We compared the effectiveness of interactions between able-bodied individuals and our robot, by comparing these two different scanning interfaces, by performing a simple task which is the grasping of a cup among other cups that are arranged in a row. The experiments were conducted using the Baxter robot [2].

The input device connected to both of our user interfaces could be any of the known access switches available in the market such as single-switches, lip switches, ball switches, box switches etc. Thus, our system is generic and allows the integration of different sensing devices.

II. RELATED WORK

Over the last years there has been extensive research for controlling a robot by disabled people either by voice or by mind. Hochberg et al. [7] have conducted research towards neurally controlled robots. This have provided people with paralysis the ability to control robotic arms using brain-computer interfaces. However, research in this field is still at its early stages and the current technology is crude. Also, because this method requires electrodes to transmit the signals, either the electrodes have to be attached outside of the skull with the disadvantage of detecting very few electric signals from the brain, or electrodes have to be placed inside the skull, thus creating scar tissue in the brain.

Chen et al. [8] have also conducted research toward this field. Willow Garage's Robots for Humanity is a project which tries to explore ways for quadriplegic people to use a PR2 robot [9] as their surrogate. Disabled people are able to use a head tracker in order to operate a variety of experimental user interfaces. These interfaces allow them to directly move the robot's body, including its arms and head. They also let them invoke autonomous actions, such as navigating in a room and reaching out to a location. Although

[1]Computer Science Department - Rutgers, the State University of New Jersey, NJ, USA. firstname.lastname@cs.rutgers.edu

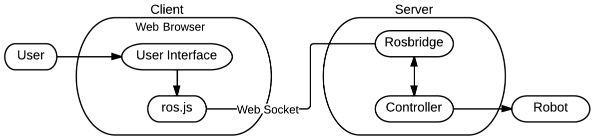

Fig. 1. System Architecture: On the left side there is the user and our client-side, whereas on the right side there is the robot and our server-side.

this effort has been successful, it is limited by the fact that the user has to be able to control a cursor on-screen through a headtracker.

More recently, Wakita et al. [4] developed a user interface with single switch scanning in order to support quadriplegic people. Although this system moves a robotic arm with a single-switch scanning interface, it requires the end-user to control all degrees of freedom of the robotic arm independently. However, visualizing the motion of the robot in terms of its joint movements is difficult for robots which are not anthropomorphic. There are indications that the end-effector is an important representative of robotic motion for high dimensional manipulators [13]. So, an interface that lets the user directly control the end effector in a 3D workspace, instead of joint positions in a high dimensional configuration space, proves more intuitive. In this work, we will show, that these kinds of interfaces are not as usable as an interface that controls only the end-effector of the robotic arm.

Regarding the web perspective of human-robotic applications, there are no previous efforts to the best of the authors' knowledge combining both a web-based robotic interface and a single-switch scanning interface designed for small devices.

III. System Architecture

In order to develop both of our systems we used the (ROS) Robot Operating System [3] which is a sophisticated open source robot middleware platform which provides hardware abstraction, device drivers, and many useful libraries and tools to develop robotic applications. The architecture of the system we adopted is depicted in Figure 1. On the left side there is the user and our client-side, whereas on the right side there is the robot and our server-side. The communication between the robot and the server side is done via a controller which communicates with the robot and an application layer network protocol, the rosbridge [6]. On the client side, which is actually the web-browser, resides our user interface which communicates with the ROSjs [5], which is a javascript library that manages connections to the rosbridge over HTML5 websockets. It provides a simple way to handle topic publishing/subscribing and services using serialized JSON objects. Then, the user interface communicates with rosbridge server via ROSjs. On server side, rosbridge listens to the clients requests and process them in ROS which in the end commands the robot to execute actions.

The two basic components of our system are the user interface component and the controller component.

A. User Interface

For the purpose of our work we created two different user interfaces, one for the direct control of all the joints independently and one for the control of the end-effector. Both of the user interfaces contain a main panel and a control panel. The main panel supports the core functionality of our system which is the control of the robotic arm. It consists of a hierarchical set of blocks, each one performing a different functionality. On the other hand, the control panel contains information concerning the settings of the system. Next, we will analyze the parts of our user interfaces.

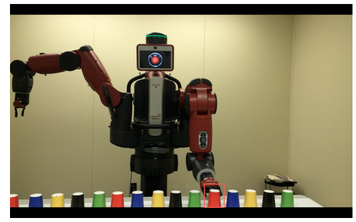

Fig. 2. Baxter robot grasping the orange cup

1) Direct manipulation of all the joints of the robotic arm: The first user interface is an interface with which the user will be able to handle all the joints of the robotic arm separately. This means that the user is able to handle the shoulder of the robotic arm, the elbow, the wrist and the end effector, one of them each time. It consists of four main blocks in the first level which contain information about the movement of the separate joints of the robotic arm. Each block is highlighted sequentially and during that time the user will be able to select one of them by using the single-switch. When one block is selected, then the elements which are inside that block start to be highlighted until the user presses the switch button again and then the robotic arm is moved according to that command (figure (3)).

The blocks of the user interface are four in number and are responsible for the movement of the wrist, the elbow, the shoulder, the base and the gripper. The first three blocks (for the wrist, elbow, and shoulder) contain another four elements inside them which are used for the translation (up and down) and the rotation (clockwise and counterclockwise) of the corresponding robotic part of the arm. The last block contains information about the movement of both the base and the end effector of the robotic arm. The base can only rotate, thus we have two elements for that purpose, and another two elements that command the robot to open or close its gripper.

Note that the blocks contain five elements. The fifth element is the return button that gives the user the opportunity to return from the element scanning of the selected block to the scanning of the blocks again.

2) Control of the end-effector: The second user interface is the interface that contains information only for the movement of the end effector of the robotic arm, depicted in figure 4. In this case, the user controls 8 blocks. The first six of them are responsible for the translation of the end effector to six different directions which are: forward, backward, upward, downward, left and right. The other two blocks are responsible for the opening/closing of the gripper.

As mentioned previously, except for the main panel, both of the user interfaces contain a control panel. The control panel is the red bar on the upper part of the screen depicted

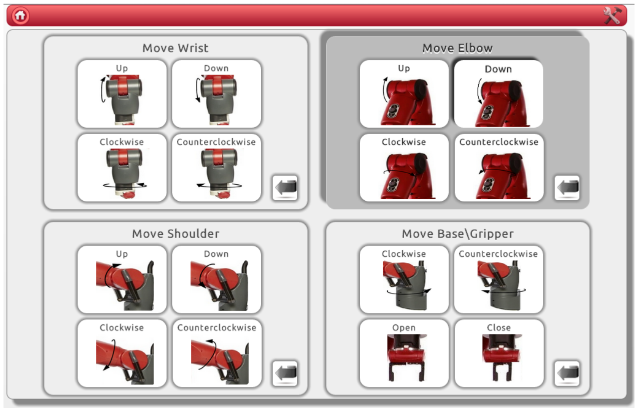

Fig. 3. User interface for controlling all the joints of the Baxter robot.

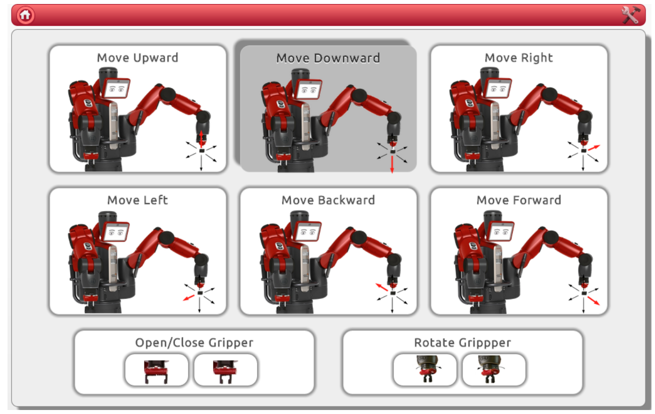

Fig. 4. User interface for controlling the end-effector of the Baxter robot.

in figures 3,4. By using this panel, the user can alter the parameters of the user interface, such as the scanning time and the order of the elements inside the blocks.

The user interfaces were implemented using HTML5, CSS3, javascript and jquery-mobile.

For the first user interface that implements the direct manipulation of the robotic arm the controller component is implemented in a Python server-side script. This script commands the robot to do certain actions. Because the robotic arm of our baxter robot has 7 degrees of freedom the actions are sixteen in total, including the closing and opening of the gripper. These actions are:

- *Set robotic arm to start position.* The start position is a specific configuration we assigned for the robotic arm.
- *Move a particular joint.* The user commands the robot to move one of its joints.
- *Move a particular joint with a specific direction.* When the user selects the joint to be moved, s/he selects also the direction s/he wants it to be moved, which can be one of up/down or clockwise/counterclockwise.
- *Open/ Close the gripper.* The robot opens/closes its end effector to grasp an object.

B. Controller Component

Regarding the second user interface that supports only the control of the end effector, the controller component contains the implementation of a motion planning algorithm. In order for the user to control only the 3D position of the end-effector, we had to incorporate inverse kinematics so as to resolve the configuration of the remaining arm. The motion planning algorithm that we used was an implementation of the Open Motion Planning Library (OMPL) [11] and the motion planning method used was PRM* (Probabilistic Roadmap Method) [12] .

C. Communication Protocol

To realize the aforementioned functions in the controller component, we needed a communication protocol between the controller and the interface. The output data of the first user interface component are an id that characterizes uniquely each joint of the robot arm as well as a sign that corresponds to the direction that the user wants to move the selected joint. The robot component receives the above data and moves the arms along with them with a predifined velocity that we have defined. The output data of the second user interface, is only an id that characterizes uniquely the direction that the user wants to move the end-effector. According to that id, the controller moves the end-effector 2 cm to the direction assigned by the user, based on its previous configuration.

IV. Experimental Setup

The overall experiment consisted of one specific task conducted using both user-interfaces. The purpose of the task was to grasp an orange cup that existed among others on a table, with a specific manner, as it is depicted in figure 2.

The experiments were performed with four fully able people that were provided a single-switch interface. The steps that we followed were the following:

1) We introduced the subject to the experiment, the robot and the user interface.
2) We let the subject, during a training phase, to become familiar both with the user interface and the robot.
3) We kept two different recordings for the interaction. One for the movement of the robot, and one for the facial expressions of the subject being examined.
4) We allowed the subject to complete all the tasks without any time limitation.
5) We administered a questionnaire to the subject.

A. Metrics

The metrics that we measured are based on the study of Kidd et al. [10]. The metrics that were appropriate for our study were:

1) Mission effectiveness: This metric is actually a boolean value that indicates if the task assigned to our subject is executed or not. We assumed that the task was completed effectively if it was executed in less than 10 minutes.

2) Mission completion time: This metric indicates the total completion time of the task given.

3) Number of collisions: This metric indicates the number of collisions of the robotic arm either to objects in the scene, (e.g. the table), or to the neighboring cups of our target cup.

TABLE I

QUANTITATIVE DATA

	Direct control of all joints				End effector control			
	Subject 1	Subject 2	Subject 3	Subject4	Subject 1	Subject 2	Subject 3	Subject 4
Mission effectiveness	NO	YES	YES	YES	YES	NO	YES	YES
Total completion time(min)	8	10	6:30	8	3:30	22	3	4
Number of collisions	2	1	0	1	0	3	0	0

4) Human behavior - Behavioral measures: Behavioral measures include data gathered from our subjects' activities during the experiment, by using the video tape. This includes mostly our subjects' comments and facial expressions.

5) Human behavior-"Self-report" measures: Self-report measures are actually self-report questionnaires. What is measured here, is the satisfaction of each one method proposed.

B. Data Collection

After the experiment the data that we collected were both qualitative and quantitative. The qualitative data include the post-experiment questionnaire administered at the end of each user session and the observer's personal notes that include the behavioral measures. The quantitive data include the total run time of the task asked, from the time we prompted our subject to do the task until the robot ended up at his goal position, as well as if our subject managed to execute the task given successfully within the time restriction and finally the number of collisions of the robotic arm.

The results for the quantitative metrics are shown in Table 1. We can observe that the most straightforward and easy to communicate interface is the interface which controls only the 3D position of the end-effector. As we can observe for the data in the table our subjects needed less time to execute the task asked and also the number of collisions with the objects in the environment were less. However, it is worth mentioning that there was one subject that needed 22 min to execute the task with this user interface. This subject during the experiment was trying actually to move the end-effector in order to control the joints of the robotic arm and bring them in such a way so as to resemble a human movement.

With regards to the questionnaires administered to the users, the results were the same[1]. The subjects mentioned that the most useful user interface was the one for the direct control of the end-effector. They mentioned that it was easier to learn to operate it, because it had fewer controls and it was closer to the human perception. Nevertheless, this user interface is not robust. The robot was not always responding correctly and some times it needed much time in order to find the path to follow in order to execute the action asked. In addition, when the arm was close to its joint limits, small steps caused large configuration changes. This gave rise to unintuitive movements. On the other hand, the subject that preferred the other user interface mentioned that this interface gave her the freedom to move each part of the robot independently and it was easier to visualize how much, where and with which order she wanted to move the joints. Nonetheless, it was hard to remember the association between user interface and the robot components and it was also hard to find the sequence of movements of the joints in order to execute a specific action.

[1]The questionnaire questions and the comments of our subjects can be accessed from <http://goo.gl/vYcGHl>.

V. CONCLUSION

In this paper, we presented the development and comparison of two different web-based robotic applications with single-switch scanning interfaces intended for tablet devices for quadriplegic people. The two interfaces consisted of a set of blocks containing either commands for direct manipulation of all the joints of a robot, or control of the end effector. As a preliminary study so as to guide the further development of appropriate interfaces for this purpose, our interfaces were tested by able-bodied subjects. Most of them stated that an interface controlling the 3D end-effector is easier than one controlling individual joints. Currently the end-effector centric interface is based on a downward pointing orientation for the end-effector. Future work may include giving this orientation control to the user. Evaluation of the interface with the target group of motor-impaired individuals is a future direction of work.

REFERENCES

[1] C. E. Steriadis and P. Constantinou, Designing human-computer interaction for quadriplegic people, ACM Trans. Computer-Human Interaction, vol. 10, pp.87-118, 2003.

[2] Baxter robot, http://www.rethinkrobotics.com/products/baxter/

[3] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, ROS: an open-source robot operating system, in Proc. Open-Source Software workshop of the International Conference on Robotics and Automation, 2009.

[4] Y. Wakita, N. Yamanobe, K. Nagata, and E. Ono, Single-Switch User Interface for Robot Arm to Help Disabled People Using RT-Middleware, Journal of Robotics, Article ID 528425, 13 pages, 2011.

[5] S. Osentoski, G. Jay, C. Crick, B. Pitzer, C. DuHadway, and O. Jenkins, Robots as web services: Reproducible experimentation and application development using rosjs,in ICRA, 2011.

[6] C. Crick, G. T. Jay, S. Osentoski, B. Pitzer, and O. C. Jenkins, Rosbridge: Ros for non-ros users, in Internat. Symposium on Robotics Research (ISRR 2011), Flagstaff, AZ, USA, August 2011.

[7] L.R. Hochberg, D. Bacher et al. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm, 2012: 372-375.

[8] T. L. Chen, M. Ciocarlie et al. Robots for humanity: A case study in assistive mobile manipulation. IEEE Robotics and Automation Magazine, Special issue on Assistive Robotics, 20.

[9] http://www.willowgarage.com/pages/pr2/overview

[10] Kidd, Cory D., C. Breazeal. "Human-robot interaction experiments: Lessons learned." In Proceeding of AISB, vol. 5, pp. 141-142. 2005.

[11] Ioan A. ucan, Mark Moll, Lydia E. Kavraki, The Open Motion Planning Library, IEEE Robotics and Automation Magazine, 19(4):7282, December 2012. DOI: 10.1109/MRA.2012.2205651.

[12] S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning, The Intern. Journal of Robotics Research, 846-894, 2011.

[13] M. Zhao, R. Shome, I. Yochelson, K. E. Bekris, E. Kowler, An Experimental Study for Identifying Features of Legible Manipulator Paths, Intern. Symposium on Experimental Robotics, Morocco, 2014.