Robot Arm Telepresence for Impaired Users Using Gaze and Speech Recognition

Alexander Adolfsson[1,2] **Digby Chappell**[1,3] **Annalaura Lerede**[1,3] **Alina-Irina Serban**[1,3] **Timo Thans**[1,2]

{aa8619, dec19, al918, ais115, tt719}@ic.ac.uk
1. Equal contribution
2. Department of Electrical and Electronic Engineering, Imperial College London
3. Department of Computing, Imperial College London

***Abstract*—People suffering from quadriplegia are dependent on a caregiver to help them with the simplest daily activities. Technological developments can help reduce the stress and burden of the caregivers and of the people suffering from quadriplegia as well as improve the quality of life of the user and give them a sense of individuality. Robotic Arms are increasingly popular to help perform simple daily tasks and alongside remote control options that use gaze and speech recognition, they become a suitable tool for people with full body paralysis. The focus of this project is to provide a telepresence system that can be solely used by a person with quadriplegia just through their eyes and speech commands for simple interaction tasks. This report describes the project hypothesis, related work, design, results, discussion, future work and conclusions.**

Github repository: https://github.com/dchappell2203/human-centered-robotics

I. Introduction

Nowadays 50000 people live with spinal cord injuries in the UK and 2500 new patients are diagnosed with it every year [1]. An increase of the 60% in the number of people injured per year was recorded compared to what was previously believed by the NHS itself. Over half of these injuries affect the cervical nerves [2] leaving the patients almost entirely or completely paralysed from the neck down. This condition is called quadriplegia and the higher the cervical injury, the more severe is the loss in mobility and perception (lower cervical injuries can result in some sensation and movement left). Sufferers of quadriplegia are forced by their condition to rely on caregivers to help them in every aspect of their lives, from regulating their body temperature to getting dressed, and have more chances to be affected by conditions such as pneumonia or muscle atrophy compared to healthy individuals.

Giving these people back even a tiny bit of autonomy or a way to take part in daily activities would not only help them cope with their disease better, but also give them a way to look after themselves and be able to contribute to society. Giving them the chance to have a job and some sort of normality in their life would relieve them from feeling like a constant weight for their caregivers and loved ones. One way to achieve this is through Robot Arm telepresence: the patients could remotely control a Robot Arm through vocal commands and gaze while in the commodity of their bed wearing a Virtual Reality (VR) headset that would give them a view of the arm while tracking their gaze and recording their voice. The idea is to utilise what is left of their mobility and augment it to achieve much more. By having the patients wear a VR headset and live streaming in the VR the images recorded through a webcam located where the Robot Arm is, they could actively participate and interact with an external reality without needing any help, having the impression of being there (a level of telepresence can be achieved).

II. Hypothesis

This study aims to provide a proof of concept on the suitability and potential of the Robot Arm Telepresence to fulfil these aims, by developing and deploying the system for some basic tasks. In particular, the following hypotheses will be investigated:

- *Intention detection*. Robot Arm Telepresence will be able to move its gripper to any point on a plane by detecting where the user gaze is looking at on that plane.
- *Command detection*. Robot Arm Telepresence will be able to recognise a set of vocal commands to execute three different tasks: calibrate, open and close.
- *Task execution*. Robot Arm Telepresence will allow the user to pick/drop objects of different widths from/to specific locations.
- *Calibration requirement*. Robot Arm Telepresence will need to be calibrated at the beginning of each use case and for each different user (not during usage).
- *Human factors*. Robot Arm Telepresence will not cause any kind of sickness or negative feelings in the user due to continued use.

III. Related Work

A. Telepresence

Telepresence robots allow users to appear to be present, feel like they are present or have some effect in a space the person does not physically inhabit [3]. In general robotics, this concept has had successful applications in allowing human interaction with hostile environments, for example in space-robotics [4]. In contrast, in a healthcare setting, telepresence

typically takes one of two forms; either a patient uses telepresence to interact with the world, usually to combat social isolation (as in [5]), or an external user uses telepresence to interact with the patient (see Fig. 1). The majority of research into healthcare telepresence has focused on the latter, as operating a telepresence robot is a complex, involved task. For example, [6] explored a telepresence robot operated by dementia patients' families in order for the patients to feel social presence; arguably providing telepresence in the 'wrong' direction, but done so because the patients in question were unable to operate the robot.

Fig. 1. Assistive telepresence robot for elderly healthcare [7]

B. Human Robot Interaction for Impaired Users

As mentioned, a large amount of telepresence research has been conducted with the aim of having able-bodied users operating the robot. In this project's case, focus is instead given to enabling severely impaired users to use telepresence as a means of interacting with the world in a way otherwise unavailable to them. Eye tracking has been researched extensively in literature, with attention recently being given to non-invasive eye tracking methods for disabled users [8], [9]. Eye tracking technology combined with a virtual reality headset, as in the HTC Vive Pro Eye [10], has the potential to create an immersive telepresence platform with non-invasive user interaction suitable for impaired users. Eye tracking-based control, also known as gaze control has been used effectively for telepresence, particularly mobile robots [11], [12], but interaction complexity is limited with just inputs from the eyes.

Alternative human robot interaction methods from the human head can add further complexity to the level of control able to be achieved. More invasive methods, such as cheek activated switches - notably used by Stephen Hawking [13], can provide a binary input that is far different from the location-based gaze input provided by eye tracking. A non invasive alternative to this, such as facial expression detection, which has been shown to be a useful input for robot control [14], could provide a similar control input without requiring specialised sensors. Voice control, used in [15], can provide an extremely rich control input for discrete robot actions, but require a higher level of patient capability.

IV. Hardware Design

The hardware used can be categorised into two groups, the human side and the robot side. Each side has distinct hardware: the human side has the VR headset and the human computer (as shown in Figure 2) while the robot side has the robotic arm, the camera, and the robot computer. The setup of the hardware together with the communication pathways are shown in Fig. 3. The human computer runs all software used by the user, such as the gaze tracker, the voice recognition, and the VR headset. The robot computer runs the software needed for moving the robot arm and the live feed from the camera. The human side hardware and the robot side hardware are in different locations.

Fig. 2. Human side set up

A. HTC VIVE Pro Eye

The HTC VIVE Pro Eye was chosen as VR headset on the human side. This device does not only offer a better and optimised ergonomics (allowing for prolonged use) and a higher screen quality (world-class graphics) compared to other headsets in commerce, it also includes a built-in precision Tobii eye tracking technology that allows to detect what the users focus their gaze on and for how long. The headset can offer the users up to a 110 degree view on the Robot Arm through the VR transposition of the images recorded through the camera on the robot side.

B. Kinova Gen3 Robotic Arm

The Kinova Gen3 robotic arm was chosen mainly because of two reasons. The first reason is its 7 degrees of freedom, making it flexible and able to move to any position within a certain radius. The second reason is that Kinova has open

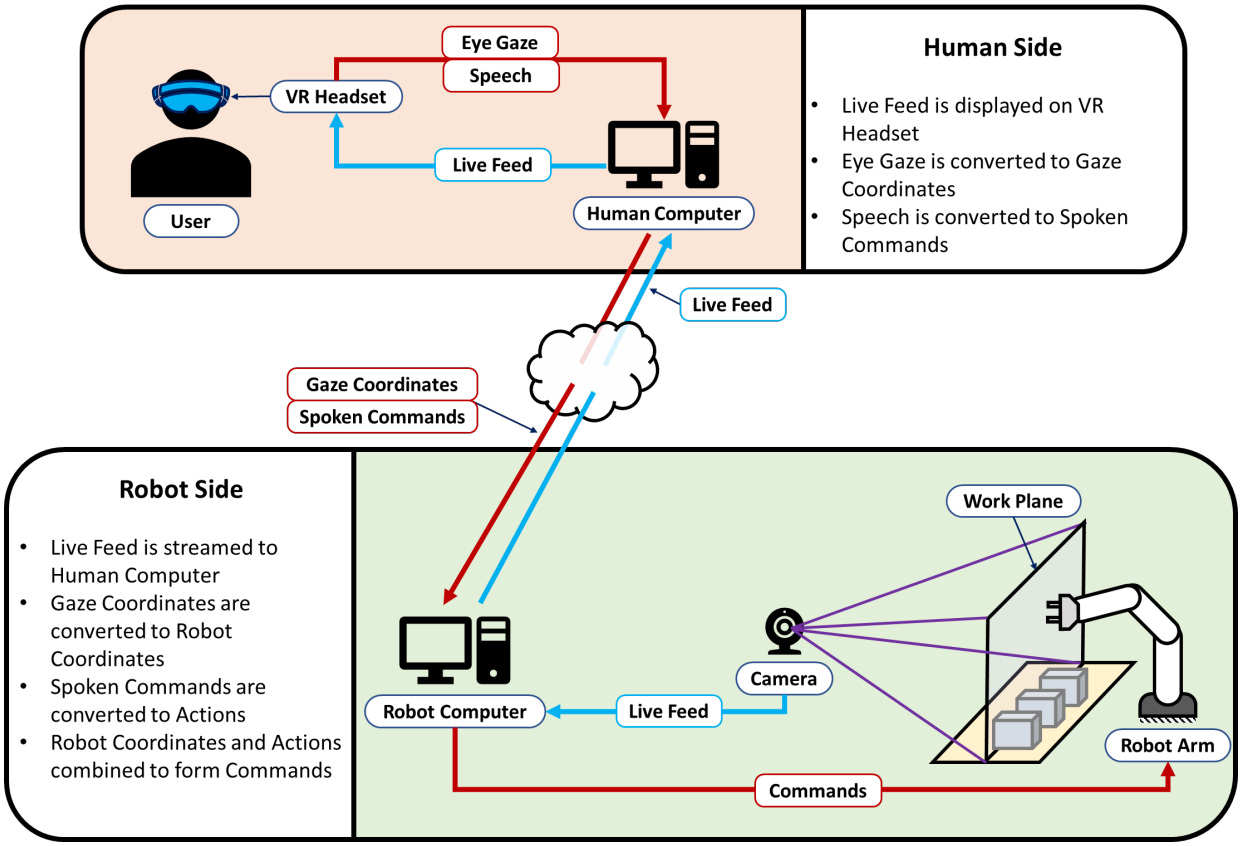

Fig. 3. Hardware diagram of the system

source driver software on Github. The arm is equipped with a Robotic 2F-85 gripper capable of grabbing objects less than 85 mm thick.

C. Camera & Robot Computer

The camera used was the laptop web camera as it's a simple way to stream the view of the robotic arm to the HTC Vive. The video feed os set to low quality to be fast enough to stream to the server and still have a decent frame rate. The web camera gives a two-dimensional view of the robotic arm and its surroundings. The laptop itself acts as the robot computer, running the robotic arm software.

V. Software Design

The software can be categorised into two groups as for the hardware: the human side and the robot side. The human side is the software designed to interact with the users (Figure 2) directly and send their inputs towards the robot computer. The robot side is the software that receives the inputs and converts them into control commands that the Robot Arm can understand.

A. Virtual Reality

Stream VR and SR_Runtime are needed on the human computer to run the eye tracking software. The calibration can then be accessed in VR through the dashboard. Unity was chosen as VR engine to display the live feed to the user on a screen and access the eye tracking data. The screen was created in Unity as a GameObject which had the live feed as texture and was set to have a fixed position relative to the user gaze origin and occupy most of the available field of view. Tobii XR SDK had to be imported into the Unity Scene to access all the eye tracking functionalities available with the headset. Through this API it was possible to access at any frame eye tracking data such as: gaze origin, gaze direction and gaze validity.

B. Camera Feed

A live camera feed is captured using a built-in laptop webcam and Xeoma video surveillance software [16]. The 320 by 240 pixel camera feed is streamed to a web server which can be accessed by Unity, where the latest camera image is loaded and displayed on the viewing screen.

C. Eye Gaze

The relevant eye gaze data accessible through the Tobii XR SDK is provided as two vectors: normalised direction of gaze and gaze origin, both given in the world reference frame. Every frame a script is set up to check whether a valid gaze can be tracked and, if it is, the gaze origin and direction vectors are stored to be converted. In order to convert these two vectors into two coordinates on the fixed screen (local reference frame with origin at the centre of the screen and moving with it),

they are first made relative to the user's head (note that the origin vector is zero relative to the user's head). The position on the screen is then evaluated as:

$$\vec{p} = \frac{Z_{screen}}{z_{gaze}} \vec{d} \tag{1}$$

A diagram illustrating this is shown in Fig. 4.

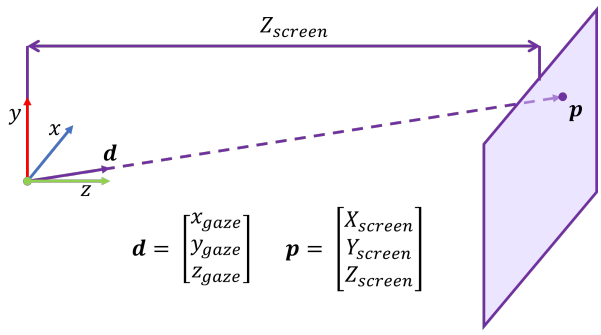

Fig. 4. Converting gaze direction and origin to gaze coordinates

The converted screen coordinates are sent from a Unity-Python TCP server running in Unity to a client running in Python. The coordinates are sent as a string in the format: `[X0.0000,Y0.0000]`, allowing the Python client to parse the string and easily split into the individual x and y screen coordinates.

D. Speech Recognition

Speech recognition is chosen as a means to lock the choice of the user by using simple commands such as: "calibrate", "open" and "close" to control the robotic arm. The operation of the speech recognition works in cooperation with the gaze tracking such that the user has to position their gaze on the object of interest before using the commands.

The development of the speech recognition module uses Python's speech recognition package alongside Google Web speech recognition library. The source microphone is integrated within the Vive Pro headset. The module listens to the user and recognises the spoken words. To eliminate noise interference the speech recognition package integrates a noise sensitivity function which is adjusted to a favourable value depending on the level of the background noise. The data gathered from the user as such is then threaded into the full system communication and the details are communicated to the robot's computer.

E. Human-Robot Communication

Communication between the human computer and robot computer is achieved by running a Human-Robot TCP server using Python on the human computer. The Python script receives incoming gaze coordinates from the Unity-Python server, and sends it to the Human-Robot client. A separate thread runs to continuously listen for speech commands, which are sent to the Human-Robot client when identified.

The robot computer client listens for new data, differentiating between gaze coordinates and speech commands by checking the incoming data type. Gaze coordinates and speech commands are published on corresponding rostopics, ready for the robot arm controller to subscribe to.

F. Arm Controller

The Kinova Gen3 robotic arm is controlled with ROS using the driver from Kinovas Github from an external computer. The arm operates under Cartesian coordinate system originated from the base. The objects that the arm can interact with have to be in a plane that is perpendicular to the web camera, in front of the robotic arm. The controller gets two type of inputs from the user, the gaze coordinates and the voice command. The gaze input is the x and y position on the plane which the arm can interact with. Because of the the inaccuracy of the eye gaze and other limitations the coordinate input is modified. The value is rounded to the closest integer to remove unnecessary decimals. There is also a limit on the y direction to remove negative values. This limit is since the object picked up by the robotic arm is located on a table that is the same height as the base of the arm.

The second input is the voice command which is used to perform the actions. The possible action the robotic arm can is pick an object or place an object at the coordinates recorded by the eye gaze. The arm moves behind the plane between picking or placing object as to not collide with any object. Because of limitations in the software for the gripper, all objects that are to be used need to be measured. The gripper is unable to adjust its grip width by itself. All object therefore need to have the same or similar width otherwise it would not be able to pick up the object. Another limitation is that the gripper has been programmed to only close horizontally.

G. calibration

To get the true coordinates from the eye gaze coordinates the system has to be calibrated. This is a typical digital image processing problem and the affine transform has to be used. The general affine transformation is commonly written as shown below [17]:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = M \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{2}$$

There are three unknown variables so the user will have to calibrate for three points. This works as the following:

- Calibration initialisation
- Robot arm moves to pre-programmed coordinate
- User fixates eye gaze on the gripper of the robot arm
- User says 'calibrate'
- Pixel coordinates are stored
- Repeat three times.

The transformation matrix 'M' is calculated in the following lines of code:

```
self.M = cv2.getAffineTransform(self.eye_crs,
rob_crs)
coordinates = np.matmul(self.M, np.array([self.
x_pixel,self.y_pixel,1]).T)
```

Listing 1. Affine Transform

The function *getAffineTransform()* from the cv2 class simply returns the transformation matrix when three points are given as input.

VI. Experimental analysis

A. Aim

The aim of the experiments carried out was to test the hypothesis above stated. In particular, to assess the usability, reliability and accuracy of the system in performing some basic tasks.

B. Testing protocol

Ten users within an age group of 19 to 25 years old were asked to try the Robot Arm telepresence and provide feedback on it. Each individual user was provided with a testing protocol with a detailed explanation of the tasks to complete and on how to provide inputs to the system. This included the list of vocal commands available. The calibration was performed for every user and they were asked to communicate when and if they felt the need to calibrate the system again. As a first task, the users were simply asked to get acquainted to the system and try and move the Robot Arm to a specific point on the work plan (to get familiar with the gaze and vocal commands). For the second task, the users were divided into two sub-groups and asked to pick an object from a specific location and place it back in the same spot. One sub-group had to use a 6cm wide object, the other one a 5.3cm wide object. The users had to repeat the second task five times for four different locations. Each location corresponded to a different intensity level, based on the angle of view and position of the object under test compared to the centre of the screen. At the end of the second task, the users were asked to fill in a survey (Appendix A) to provide feedback on the system and rate their experience. The survey is split into four different sections:

- Evaluation of the VR headset
- Evaluation of the speech recognition
- Evaluation of the entire system
- Miscellaneous questions

This allows the user to provide a detailed assessment of the system and its components.

C. Data collection

The following data was collected for each user:

- number of total successes in picking up the same object out of total number of trials
- number of total successes in picking up the same object from a certain location out of total number of trials for that location
- error distance (away from ideal location) done placing the object back averaged over all trials
- error distance (away from ideal location) done placing the object back averaged over all trials for each specific location
- number of successful detections of vocal commands out of total number of attempts
- number of successful detection of each specific vocal command out of total number of attempts for that vocal command
- user-based feedback
- number of requests for re-calibration

D. Outcome measures

The outcome of the experimental testing was categorised in three main measurements: the gaze accuracy, the speech accuracy and the user experience which was evaluated with the help of a survey as provided in Appendix A.

The gaze accuracy was determined by measuring the error distance from the guidelines provided (from the ideal location of each level) for every pick and place action. The set-up is shown in Figure 5.

Fig. 5. Test set-up with different intensity levels

The speech accuracy was recorded manually while the user was performing the task. The evaluation was based on a number of attempts count.

Lastly the feedback survey provided an evaluation of the system from the user's perspective allowing the developers to improve it accordingly.

E. Results

Each testing session took approximately 40 minutes to complete per user. The data collected is summarised in Table I and II. These show that the rate of success in managing to pick up an object increases with decreasing thickness of the object. This can be deducted by Figure 6: the success count is higher for the sub-group 2 for three out of four levels. Considering the first object is almost as wide as the grip of the robot arm, it requires a more accurate control to manage to pick this up, compared to the thinner object. As the level of attention of the users quickly drops over the trials while their fatigue increases, their performance on average gets worse with time. Figure 6 also shows that it is easier for the users to pick object

	level 1		level 2		level 3		level 4		total		speech success		
user	pick	place [cm]	pick	place [cm]	pick	place [cm]	pick	place [cm]	pick	place [cm]	close	open	tot
1	4/5	3.4	2/5	2.68	1/5	2.5	2/5	3.14	9/20	2.93	20/37	20/42	40/79
2	4/5	4.1	1/5	3.46	2/5	1	1/5	1.82	8/20	2.595	20/38	20/28	40/66
3	3/5	2.9	2/5	2.38	3/5	1.73	1/5	2.55	9/20	2.39	20/29	20/31	40/60
4	3/5	3.62	2/5	2.84	2/5	2.5	0/5	2.12	7/20	2.77	20/41	20/35	40/76
5	4/5	3.15	3/5	2.22	3/5	1.56	2/5	2.38	12/20	2.33	20/33	20/28	40/61
									45/100	2.603	20/35.6	20/32.8	40/68.4

TABLE I
DATA COLLECT FOR SUB-GROUP 1 GIVEN THE WIDER OBJECT (6CM).

	level 1		level 2		level 3		level 4		total		speech success		
user	pick	place [cm]	pick	place [cm]	pick	place [cm]	pick	place [cm]	pick	place [cm]	close	open	tot
6	5/5	1.52	4/5	0.76	2/5	0.78	3/5	1.58	14/20	1.16	20/25	20/36	40/61
7	5/5	0.88	1/5	2.22	2/5	3.56	0/5	2.18	8/20	2.21	20/45	20/40	40/85
8	5/5	0.82	3/5	1.64	0/5	2.1	2/5	1.44	8/20	1.5	20/36	20/42	40/78
9	5/5	1.44	5/5	0.98	3/5	1.32	2/5	1.68	15/20	1.355	20/29	20/34	40/63
10	4/5	0.59	4/5	0.93	3/5	1.24	2/5	1.17	14/20	0.9825	20/37	20/41	40/78
									59/100	1.4415	20/34.4	20/38.6	40/73

TABLE II
DATA COLLECT FOR SUB-GROUP 2 GIVEN THE THINNER OBJECT (5.3CM).

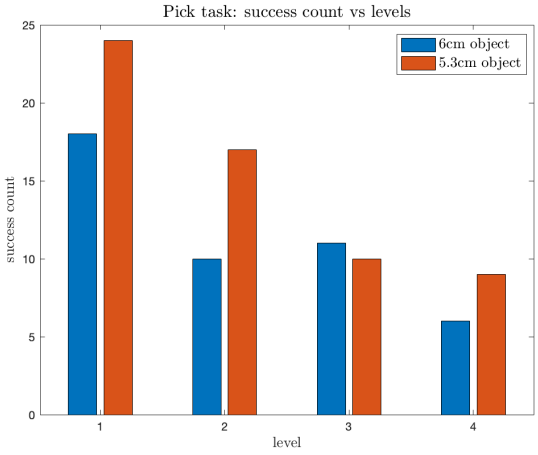

Fig. 6. Success count for pick task for the two sub-groups

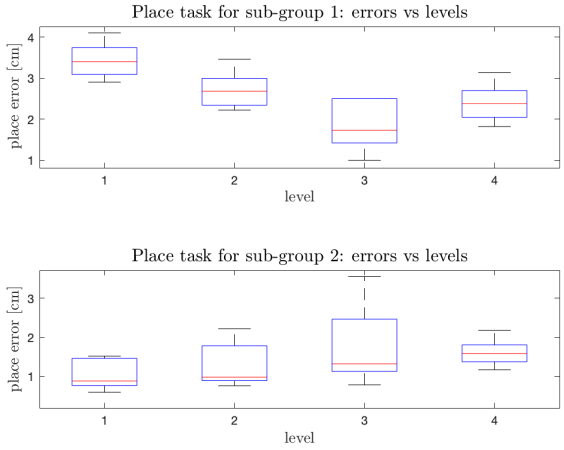

Fig. 7. Placement error distances across users for different levels

that are at the centre of their field of view: the success count on average drops with increasing level of difficulty, i.e. with increasing angle of view.

For what concerns the placing tasks, sub-group 1 commits on average an error of 2.6cm in placing the object back in its ideal location. Sub-group 2, instead, has an average error of 1.44cm. Figure 7 confirms that the errors are considerably lower for sub-group 2 compared to sub-group 1 across all levels. Again, the accuracy in placing is increases with decreasing object width and is overall good. From Figure 7 it can be seen how for sub-group 1 the error initially decreases with increasing level difficulty, but in the end increases. This is due to the presence of two opposite trend that counter-balance each other: the users tend to improve with practice (they get more acquainted with the system), but the task gets harder (increasing the angle of view with the level). The first trend seems to dominate in the first three levels, but the second trend takes over where the difficulty is considerably higher (level 4). For what concerns sub-group 2, Figure 7 shows that the error slightly increases with increasing difficulty. This may be a sign that, for thinner objects the difficulty trend takes over the improvement trend sooner.

The speech recognition was found to be accurate in the 56.6% of the cases as can be noticed in Figure 8. More detailed success rates can be found in Table I and II. Higher accuracy values have been observed for the users that speak English as a first language (users 3, 5 and 6). This means that the used speech recognition algorithm is sensitive to accents. There is a timestamp for the speech recognition module to listen

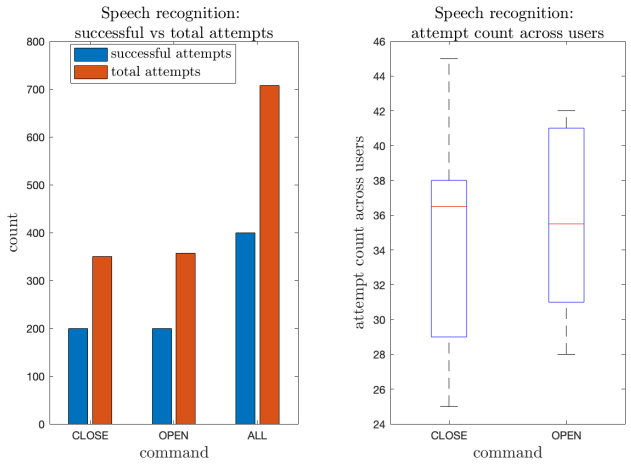

Fig. 8. Speech detection successful vs total attempts both different commands and for all commands together. Analysis considering all users variability on the left and across user variability on the right.

and recognise the phrase of the user, such that when the user pronounces the key commands outside the listening window, the system fails to register the command. This increases the time required to recognise the command, thus the time to complete the tasks and yields lower success rates. Overall the system is able to recognise the selected commands to execute the actions, although it may take more than one attempt. Figure 8 shows how the detection of the command "close" was on average and across users easier for the algorithm compared to the detection of the command "open". This resulted because the place task was always performed right after the pick task (for the way the testing protocol was set up): the listening window for the "open" command was delayed by the completion of the "close" command.

It has been observed that the re-calibration of the system for each individual user increases the accuracy of the overall performance. None of the users asked to re-calibrate the eye tracking during the performance of the task.

VII. Discussion

A. System Performance and Stability

Eye-tracking calibration was a built-in feature to the HTC Vive Pro Eye and it could, therefore, be completed at any point during the simulation. None of the users asked to re-calibrate the headset while performing the required tasks meaning that the calibration was generally robust to headset movement and to time for each individual user. Camera-robot calibration, on the other hand, was set up specifically for the designed tasks. If the camera was moved at any point during the telepresence, re-calibration was needed; however, camera-robot re-calibration generally requires a relatively short time. These evidences validate the calibration requirements stated in the hypothesis. The speech recognition is reliable for 56.6% of the cases, the results being dependent on the timing at which the user is pronouncing the key commands and the accent of every individual. The speech recognition has time frames for which the system listens, and if a window is missed, the user has to repeat the command. This can be improved by continuously listening to the user and only picking the key commands when two conditions are fulfilled: the gaze is fixed on a point of interest and the key command is pronounced. In addition to this, the virtual reality simulation was only stable for a limited amount of time. This is due to a lack of memory management; each frame loaded from the camera feed remained loaded until the simulation was terminated or ran out of memory.

B. Usability and Task Performance

The system was shown to be become less usable at larger axial offsets (higher difficulty levels), as shown in Fig. 6. This was reduced when thinner objects were used, because the open gripper of the robot was able to accommodate for errors. Placing, however, was shown to vary independently of difficulty level (see Fig. 7). It is thought that this is caused by two factors; one, there is no reference location for the user to look at, and two, there was no accommodation from the hardware as in the picking task.

In order to be usable in real life scenarios, task performance would have to be improved unless the object to grip was reasonably small.

VIII. Future Work

Memory management can be drastically improved by creating a script that automatically unloads any camera feed images that aren't in use. Unity provide a 'Resources' toolkit that can be used for similar problems to this, with functions such as `Resources.UnloadUnusedAssets();` that could be ideal. Further stability improvements are also possible such as automatic calibration and improved voice recognition.

To improve task performance at large offsets from the camera axis, the camera could be positioned further from the work space - the angular error at large offset positions would be greatly reduced, but would require a higher resolution camera.

The telepresence could be extended by introducing more complex commands such as 'rotate', or depth related commands 'closer' and 'further'. More reliable gripping of variable width objects could be achieved by automatically detecting the width of the object to be gripped from the camera feed. The core technology can be applied to other robots; for example, a mobile robot could navigate by driving to chosen locations from a screen.

IX. Conclusions

- *Intention detection*. Intention detection was achieved with good success. The robot was able to move its end-effector to any point on a plane specified by gaze.
- *Command detection* Command detection was achieved with reasonable success; in this case speed was favoured over accuracy, meaning the user had to repeat commands frequently.
- *Task execution*. Pick and place tasks were accomplished with the robot arm telepresence system, but varying widths had to be set by hand.

- *Calibration requirement* The system required calibration once at the start of use, and remained robust as long as the camera and robot remained fixed relative to each other.
- *Human factors* Continued use of the Robot Arm Telepresence system did not cause any nausea, but discomfort grew from the weight of the virtual reality headset. Because the eye gaze was only used when a spoken command was issued, the eyes could rest between commands, meaning tiredness was not a large issue.

In conclusion, this Robot Arm Telepresence system offers a suitable way for impaired users to interact with a remote environment. By implementing some features highlighted in the Future Work section, the system could be used in a real world setting.

REFERENCES

[1] Aspire. Spinal cord injury paralyses someone every four hours. https://www.aspire.org.uk/news/every-four-hours, 2019. [Online; accessed 06/11/2019].

[2] Sharon Henry. *Advanced Trauma Life Support - Student Course Manual*. American College of Surgeons, 10th edition, 2018.

[3] Techopedia. What is telepresence? https://www.techopedia.com/definition/14600/telepresence, 2019. [Online; accessed 05/11/2019].

[4] André Schiele and Gianfranco Visentin. The esa human arm exoskeleton for space robotics telepresence. In *7th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, pages 19–23, 01 2003.

[5] C. Escolano, A. Ramos Murguialday, T. Matuz, N. Birbaumer, and J. Minguez. A telepresence robotic system operated with a p300-based brain-computer interface: Initial tests with als patients. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 4476–4480, Aug 2010.

[6] Wendy Moyle, Cindy Jones, Marie Cooke, Siobhan Dwyer, Billy Sung, and Suzie Drummond. Connecting the person with dementia and family: A feasibility study of a telepresence robot. *BMC geriatrics*, 14:7, 01 2014.

[7] Saso Koceski and Natasa Koceska. Evaluation of an assistive telepresence robot for elderly healthcare. *Journal of Medical Systems*, 40(5):121, Apr 2016.

[8] T. Wu, P. Wang, Y. Lin, and C. Zhou. A robust noninvasive eye control approach for disabled people based on kinect 2.0 sensor. *IEEE Sensors Letters*, 1(4):1–4, Aug 2017.

[9] J. Park, T. Jung, and K. Yim. Implementation of an eye gaze tracking system for the disabled people. In *2015 IEEE 29th International Conference on Advanced Information Networking and Applications*, pages 904–908, March 2015.

[10] Eli Blumenthal. Htc launches eye-tracking vive pro eye in us for $1,599. https://www.cnet.com/news/htc-launches-eye-tracking-vive-pro-eye-in-us-for-1599/. [Online; accessed 06/11/2019.

[11] John Paulin Hansen, Alexandre Alapetite, Martin Thomsen, Zhongyu Wang, Katsumi Minakata, and Guangtao Zhang. Head and gaze control of a telepresence robot with an hmd. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ETRA '18, pages 82:1–82:3, New York, NY, USA, 2018. ACM.

[12] M. Minamoto, Y. Suzuki, T. Kanno, and K. Kawashima. Effect of robot operation by a camera with the eye tracking control. In *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1983–1988, Aug 2017.

[13] Jonathan Wood Stephen Hawking. My computer. http://www.hawking.org.uk/the-computer.html, 2018. [Online; accessed 06/11/2019].

[14] N. Hasegawa and Y. Takahashi. How recognition of human facial expression can be incoporated in robot control. In *2019 20th International Conference on Research and Education in Mechatronics (REM)*, pages 1–6, May 2019.

[15] Koksal Gundogdu, Sumeyye Bayrakdar, and Ibrahim Yucedag. Developing and modeling of voice control system for prosthetic robot arm in medical systems. *Journal of King Saud University - Computer and Information Sciences*, 30(2):198 – 205, 2018.

[16] Felenasoft. Xeoma - the best video surveillance software. https://felenasoft.com/xeoma/en/, 2019. [Online; accessed 17/12/2019].

[17] Toby Breckon Chris Solomon. *Fundamentals of Digital Image Processing*. John Willey & Sons, 1st edition, 2011.

VR Robot Arm Telepresence
Feedback form:

This form is split into 4 categories: A) Evaluation of the VR headset
B) Evaluation of the speech recognition
C) Evaluation of the entire system
D) Miscellaneous

A) Evaluation of the VR headset:
The evaluation is done with the help of a scale from 0 to 4 where 4 is the highest score and 0 is the lowest score.

1) Rate the level of comfort of the headset.
0 – very poor 1 – poor 2 – neutral 3 – good 4 – very good
2) Did you find it easy to calibrate the headset?
0 – very hard 1 – hard 2 – neutral 3 – easy 4 – very easy
3) What is the accuracy of the gaze tracking?
0 – very poor 1 – poor 2 – neutral 3 – good 4 – very good
4) How would you rate the precision level of the gaze tracking?
0 – very poor 1 – poor 2 – neutral 3 – good 4 – very good
5) How satisfied are you with the responsiveness of the pointer?
0 – very unsatisfied 1 – unsatisfied 2 – neutral 3 – satisfied 4 – very satisfied

B) Evaluation of the speech recognition:
The evaluation is done with the help of a scale from 0 to 4 where 4 is the highest score and 0 is the lowest score.

1) Rate the level of accuracy of the speech recognition.
0 – very poor 1 – poor 2 – neutral 3 – good 4 – very good
2) Rate the speed of the speech recognition.
0 – very poor 1 – poor 2 – neutral 3 – good 4 – very good
3) How did you find the commands to action relation? (are they good representatives of the action)
0 – very poor 1 – poor 2 – neutral 3 – good 4 – very good
4) What other commands would you like to be able to use? (write your suggestions in the box below)

C) Evaluation of the entire system
The evaluation is done with the help of a scale from 0 to 4 where 4 is the highest score and 0 is the lowest score.

1) How would you rate the overall accuracy of the system?
0 – very poor 1 – poor 2 – neutral 3 – good 4 – very good
2) Rate the easiness of use of the system.
0 – very poor 1 – poor 2 – neutral 3 – good 4 – very good
3) How satisfied are you with the video streaming quality?
0 – very unsatisfied 1 – unsatisfied 2 – neutral 3 – satisfied 4 – very satisfied

APPENDIX A
USER FEEDBACK FORM

4) How satisfied are you with the robotic arm precision?
0 – very unsatisfied 1 – unsatisfied 2 – neutral 3 – satisfied 4 – very satisfied
5) How satisfied are you with the responsiveness of the system to the commands?
0 – very unsatisfied 1 – unsatisfied 2 – neutral 3 – satisfied 4 – very satisfied

D) Miscellaneous
The evaluation is done with the help of a scale from 0 to 4 where 4 is the highest score and 0 is the lowest score.

1) How satisfied are you with the functionality of the system?
0 – very unsatisfied 1 – unsatisfied 2 – neutral 3 – satisfied 4 – very satisfied
2) How likely are you to use this system?
0 – very unlikely 1 – unlikely 2 – neutral 3 – likely 4 – very likely
3) Any other comments: (write them in the box below)