Gaze-based, Context-aware Robotic System for Assisted Reaching and Grasping*

Ali Shafti[1], Pavel Orlov[1] and A. Aldo Faisal

***Abstract*— Assistive robotic systems endeavour to support those with movement disabilities, enabling them to move again and regain functionality. Main issue with these systems is the complexity of their low-level control, and how to translate this to simpler, higher level commands that are easy and intuitive for a human user to interact with. We have created a multi-modal system, consisting of different sensing, decision making and actuating modalities, to create intuitive, human-in-the-loop assistive robotics. The system takes its cue from the user's gaze, to decode their intentions and implement lower-level motion actions and achieve higher level tasks. This results in the user simply having to look at the objects of interest, for the robotic system to assist them in reaching for those objects, grasping them, and using them to interact with other objects. We present our method for 3D gaze estimation, and action grammars-based implementation of sequences of action through the robotic system. The 3D gaze estimation is evaluated with 8 subjects, showing an overall accuracy of 4.68±0.14cm. The full system is tested with 5 subjects, showing successful implementation of 100% of reach to gaze point actions and full implementation of pick and place tasks in 96%, and pick and pour tasks in 76% of cases. Finally we present a discussion on our results and what future work is needed to improve the system.**

I. Introduction

Limitations in human upper limb movements can be a result of spinal cord injuries, neurodegenerative diseases or strokes. These adversely affect a person's ability for basic activities of daily life. Robotic solutions are being devised as alternative actuators, to assist with these issues. Devices are presented in the form of exoskeletons [1], [2], prosthetics [3] and orthotics [4]. For such systems, the user's control interface is typically either residual motion (e.g. sip and puff [5]) or neural interfaces (e.g. muscle activity [6] or brain-computer interfaces [7]). These interfaces are not available to all patients, and/or require invasive procedures and long training times for the user to be adept with their use [7]–[9]. The degrees of freedom tend to exceed the available number of independent channels within the above interfaces, and therefore result either in simplified device capabilities or in difficulties in user control. We have previously presented work on eye-tracking studies [10]–[12] and the use of eye-tracking as a robot interface [13]. In this work, we present a new method for 3D gaze point estimation, and its integration with a robotic system architecture allowing real-time intention decoding, decision making, and robot-actuated reach and grasp restoration. The system implements complex tasks by combining lower-level robotic actions, that are initiated through the user's gaze, and carried on through awareness of the interaction context and human action grammars.

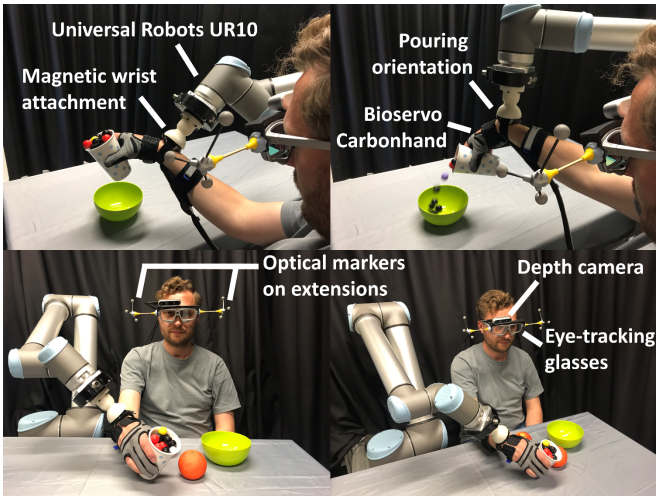

Fig. 1: A user with our robotic reach and grasp support system.

Gaze has been used successfully in the past as an interface for machines, particularly in human-computer interfaces [14] and social robotics to monitor human attention and engagement [15]–[17]. It has been implemented as an interface for robotic laparoscopic surgery [18] as well as emotion analysis [19]. When it comes to patients with movement disabilities, there is work on the use of gaze patterns in rehabilitation [20], for the control of 2 degrees of freedom in upper limb exoskeletons, where the patient uses gaze to direct the robot on a 2D surface. There is also work In assistive robotics, particularly wheelchairs controlled through gaze [21]–[24]. We aim to expand the research into 3D gaze monitoring within assistive robotics for the restoration of reaching and grasping. The use of gaze is of particular interest as it is retained in most upper limb disabilities. Furthermore, it allows for natural, easy to learn, and non-invasive interface between the human and the robot. To achieve this we rely on the idea of action grammars: our actions, like our sentences, have rules regarding how to combine them and which order to use to create a meaningful sequence.

In section II, we first present an overview of our system architecture followed by details of its consisting parts. Section III presents our evaluation experiments, with the results reported and discussed in Section IV. Section V concludes the paper, with proposals on future work.

*Research supported by eNHANCE (http://www.enhance-motion.eu) under the European Union's Horizon 2020 research and innovation programme grant agreement No. 644000.

[1]A. Shafti and P. Orlov contributed equally to this work. Along with A. A. Faisal, they are with the Brain and Behaviour Lab, Dept. of Computing and Dept. of Bioengineering, Imperial College London, SW7 2AZ, London, UK. a.shafti, p.orlov, a.faisal@imperial.ac.uk

II. Methods

A. System overview and architecture

We aim to create a robotic system which acts based on human 3D gaze patterns and the context of the environment. In order to track 3D human gaze patterns, we need to know which direction the human's eye pupils are pointed at. Commercial eye-tracking glasses provide 2D gaze monitoring without any information on depth. We use an RGBD camera mounted on top of the eye-tracker glasses to gain the missing depth information. Once this is obtained, we need the user's head position and orientation, so that we can transform the 3D gaze points obtained within the eye-tracker's coordinate system, to that of the world. This information, along with the output of the object recognition module working on top of the eye-tracking ego-centric camera images, are then to be used to make decisions and implement actions using robotic devices.

Our system consists of multiple modalities integrated and working together through the Robot Operating System (ROS) environment [25]. These include: 1. Eye-tracking glasses, 2. RGB-D camera, 3. Convolutional Neural Network for object recognition, 4. Optical head-tracking, 5. Robotic arm for reaching support and 6. Robotic glove for grasping support. The block diagram in Figure 2 depicts an overview of our system. Individual modalities are described in detail in the following.

B. Eye-tracking

For eye tracking, we use the SMI ETG 2W A (SensoMotoric Instruments Gesellschaft für innovative Sensorik mbH, Teltow, Germany). Using the SMI Software Development Kit (SDK), we are able to obtain 2D gaze positions superimposed on the ego-centric RGB camera image. To calculate 3D gaze points, we need depth information. We mounted an Intel Realsesne D435 RGB-D camera (Intel Corporation, Santa Clara, California, USA) on top of the eye-trackers, using a 3D printed frame. This can be seen in Figure 1.

Typically, eye-tracking devices should be calibrated with the user's eyes. During calibration a user has to look at several points on a physical plane, and the researcher manually marks them on the ego-centric video feed (see Figure 5). In our setup, we use the depth picture from the RGBD camera aligned to the camera's RGB image, as an ego-centric frame and map gaze points directly to it during the calibration process. The Intel API for the depth camera provides the depth of each pixel in metres.

Through this integration, we are able to obtain the Euclidean distance, d_g, between the depth camera lens, and the surface in space over which the 3D gaze point of the user sits. The camera image resolution is 1280×720, let this be referred to as $W_c \times H_c$. The 2D gaze point is sumperimposed on the same camera image, and can therefore also be referred to in terms of pixels, let this be represented as (P_x, P_y), where P_x is the horizontal gaze pixel location, and P_y is the vertical one; with the centre of the image considered the origin, i.e. $(0, 0)$ (see Figure 3). We refer to the horizontal

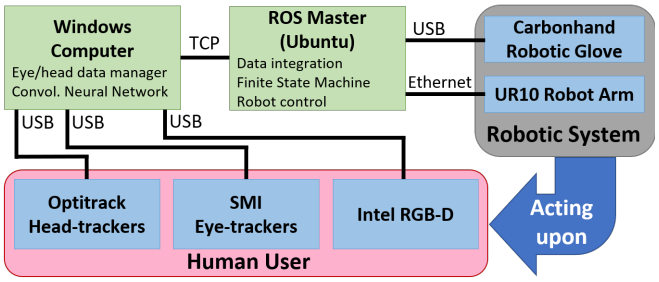

Fig. 2: Block diagram showing the architecture of our system.

and vertical field of view (FOV) angles of the camera as α_{cH} and α_{cV}, respectively. Let c be defined as the line connecting the camera to the centre of its frame. The gaze angle is then defined in two cases. Horizontal gaze angle, α_{gH}, is the angle between the projection of the Euclidean distance line above (d_g) on the horizontal (transverse) plane: d_{gH}. Similarly, the vertical gaze angle, α_{gV}, is the angle between the projection of d_g on the vertical (sagittal) plane: d_{gV}. The case of the horizontal gaze angle is displayed in Figure 3. We need to convert the gaze point values from pixels, to metres in Cartesian space: (g_x, g_y, g_z). Consider the case shown in Figure 3:

$$\begin{aligned} \tan(\alpha_{gH}) &= P_x/c \\ \tan(\alpha_{cH}) &= (W_c/2)/c \end{aligned} \tag{1}$$

Combining these we get:

$$\alpha_{gH} = \arctan(\frac{P_x}{W_c/2}\tan(\alpha_c H)) \tag{2}$$

We now know the angle between the Euclidian distance gaze line, d_g, and the centre line of the camera frame. For the vertical case, similarly, we find: $\alpha_{gV} = \arctan(\frac{P_y}{H_c/2}\tan(\alpha_{cV}))$. We know that:

$$\begin{aligned} g_x &= d_{gH}\sin(\alpha_{gH}) \\ g_y &= d_{gV}\sin(\alpha_{gH}) \end{aligned} \tag{3}$$

Also, considering the right-angled triangles formed between d_g and its projections, we can write:

$$\begin{aligned} g_x^2 + d_{gV}^2 &= d_g^2 \\ g_y^2 + d_{gH}^2 &= d_g^2 \end{aligned} \tag{4}$$

Combining equations 3 and 4 yields a system of linear equations as follows:

$$\begin{cases} \sin^2(\alpha_{gH})d_{gH}^2 + d_{gV}^2 = d_g^2 \\ d_{gH}^2 + \sin^2(\alpha_{gV})d_{gV}^2 = d_g^2 \end{cases} \tag{5}$$

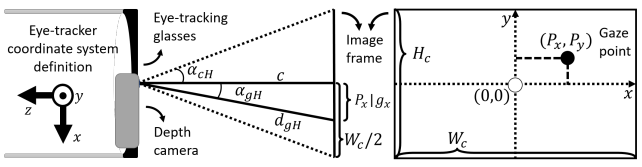

Fig. 3: Geometric representation of the gaze angle and the camera frame, used for the 3D gaze point estimation.

which we can solve to find d_{gH} and d_{gV}, and use them to obtain g_x and g_y from equation 3. Then, to obtain the z-distance of the gaze point:

$$g_z = \sqrt{d_g^2 - g_x^2 - g_y^2} \tag{6}$$

We now have the 3D gaze point of the user, in Cartesian coordinates, with respect to the defined eye-tracker coordinate system. We then need to transform this to the robot coordinate system. To do this, we need the position and orientation of the user's head.

C. Optical head tracking

We use the Optitrack Flex 13 cameras (NaturalPoint, Inc. DBA OptiTrack, Corvallis, Oregon, USA) for optical head tracking. To avoid occlusions due to the user's head, we created 3D printed extensions. This can be seen in Figure 1. The markers are then selected in the Optitrack software, Motive; and a rigid body is defined. This method gives us the position of the centre of the rigid body (i.e. the camera depth lens position) with respect to the pre-defined Optitrack origin, and the rotation of the rigid body, relative to its initial position when selected and defined in Motive. We use these values to form a transformation matrix which is applied to the 3D gaze point coordinates transforming them into the Optitrack coordinate system followed by an extra transformation matrix to transform this into the robot coordinate system.

D. Detection of objects and intention of action

We use a deep neural network approach coupled with naive classification to classify multiple objects in the user's field of view. The development of this system is highlighted in [26]. The output of this is real-time object recognition on the depth camera images, with rectangular bounding boxes drawn around the detected object. We can then use the detected gaze position on the camera image frame, along with these bounding boxes to detect: 1. which object and 2. which part of that object, is the user gazing upon. We use this information to extract context and intention - i.e. which objects is the user interested in, and whether there is an intention of physical interaction with this object. To detect the latter, we have defined the right-hand side of each object, as the location for the user to gaze at (for 15 gaze points), to indicate an intention of physical motion. This gives the user executive control, allowing them to freely inspect objects without causing robot movements.

E. Robotic system integration

Our robotic system consists of two commercial robots: 1. Universal Robots UR10 (Universal Robots A/S, Odense, Denmark) and 2. BioServo Carbonhand (Bioservo Technologies AB, Kista, Sweden). The former is used for reaching and the latter for grasping support. The user wears the robotic glove on their hand, and attaches their arm to the UR10 through a 3D printed magnetic attachment on their wrist. The magnetic setup is used to ensure our test subjects are able to detach their arm by pulling it away if they sense a risk.

TABLE I: Convention used for the categories of objects within the finite state machine.

Graspable	Pourable	GP	Comments
0	0	00	e.g. large container
0	1	01	undefined e.g. table
1	0	10	e.g. apples/oranges
1	1	11	e.g. small container

Strict workspaces, motion planning constraints including a 3D reconstruction of our lab environment and user bounding boxes for collision avoidance are in place to ensure safety. As we use ROS, the robot choice is irrelevant, as long as it is ROS-compatible.

The ROS master receives the gaze point as pixel locations within the camera frame, the object that the user's gaze falls upon and whether there is an intention of motion as well as the user's head position and orientation. The 3D gaze point calculations described above are implemented within our ROS package. We then have the user's 3D gaze point, i.e. the location of the object they are looking at, as well as knowledge on what that object is and whether the user wants to physically interact with that object. We use these inputs to make decisions and implement sequences of actions with the robotic system, using lower-level actions and following rules of action grammars. A finite state machine (FSM) is applied to implement this.

As an example for our proof of concept, we are dealing with a dining table scenario. We are therefore looking at objects such as fruits (apples, oranges) and containers (cups, bottles, bowls). We define the interaction between these objects as 1. pick and place on the table, 2. pick and place into containers and 3. pick and pour into larger containers. Grammars are already visible here, i.e. you can pick and place fruits on the table or into the bowl, but not into the cup or bottle; similarly, you cannot pour fruits, but you can pour the cup/bottle - and only into the bowl and not on the table. We categorise our scenario objects considering their graspability and pourability as defining parameters. For example, apples and oranges are graspable but not pourable, cups are graspable and pourable. We use a binary string for notation of objects. See Table I for a description. Note that a non-graspable but pourable object is undefined - this category, i.e. $GP = 01$ is used to represent the dining table itself.

The states of the FSM are defined to represent the user state. The parameters used in this definition are whether the user's grip is open or closed, and what object is held in their grip, if any. We represent this in a binary string format as well, with the grip open being represented as 0 and grip closed as 1. This is followed by the object held, using the same notation as that of Table I, except that in this case,

TABLE II: Convention used to name the states within the finite state machine.

Grip	Object held	Comments
0	01	001: Grip open, no object held.
1	01	101: Grip closed, no object (grasp failure)
1	10	110: Grip closed, graspable non-pourable
1	11	111: Grip closed, graspable pourable

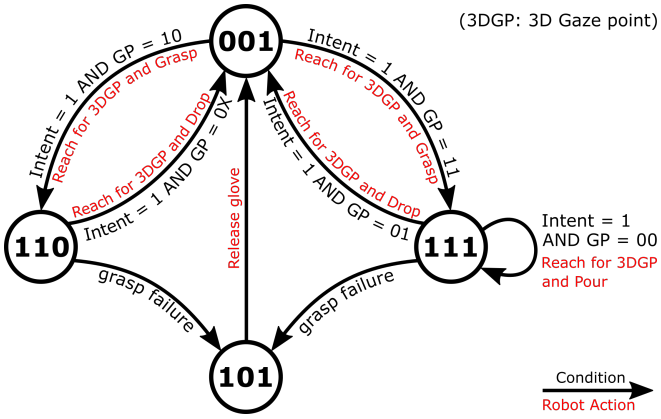

Fig. 4: The finite state machine used to implement the sequences of action.

$GP = 01$ is used to represent "no object held". This leads to a total of 4 states for the FSM, which are listed in Table II.

The full FSM, with the states, transitions, their conditions and actions, is depicted in Figure 4. The black text on each transition arrow defines the conditions for that particular transition. Note that the intention of action as detected from the user's gaze patterns, is represented as a Boolean variable here named Intent. The GP condition on the transition arrows relate to the object gazed by the user, following the convention of Table I. The red text on each transition arrow indicates the robotic action that is triggered by that transition, i.e. Reach for 3DGP and Grasp means the UR10 robotic arm will reach for the 3D gaze point of the user, and once this is completed, the Carbonhand robotic glove will close the user's grip.

The user starts in state 001 (grip open, no object held) and remains in this state as long as no intention of action is detected. Note that self-transitions in case of unfulfilled conditions are not displayed in the FSM figure for simplicity. Once the user's gaze pattern indicates an intention of action ($intent == 1$), depending on the object the user is looking at, one of the following will occur: Looking at a Graspable, non-Pourable object ($GP = 10$, e.g. apple, orange), the machine will transition to 110. The robotic system will reach for the object and grasp - and similarly for looking at $GP = 11$ (e.g. cup, bottle) it will transition to 111. If the user is looking at a non-Graspable, non-Pourable object ($GP = 00$, e.g. bowl) or the table ($GP = 01$), the machine will not transition. Note that transitions will not execute if the 3D gaze point of the user is not within the workspace, or if it is not motion planable for the robot.

The 'grasp failure' transition is to handle the potential cases when the robotic glove closes the grip but a grasp of the object of interest is not successful, or the case when a grasp has been made, but it is not stable and the object is dropped midway through the task. Failure of a grasp can only initiate from a state which involves an object having been grasped already i.e. 110 or 111. From these two states, if the grasp is unsuccessful, a transition will be made to state 101. Following the convention of Table II, 101 means that the grip is closed but no object is held. This state will immediately transition to 001 by releasing the grip. The ROS package for the Carbonhand robotic glove publishes tendon tension values, motor voltages and force sensor values from the glove finger tips. Combining these data, we are able to detect 1. whether the glove is closed or open and 2. if closed, whether the user is holding an object, or an empty grip. This is used to detect the user's state and particularly grasp failures.

Note that in practical implementation, there is a clear offset from the user's grasp point, to the robot TCP, which depends on the size and orientation of the magnetic attachment, as well as each user's particular wrist diameter, hand size and finger length (Figure 1). To personalise the system to each user, we created a calibration step: We move the robot with the user's arm attached to it to an arbitrary point on the table, at a comfortable grasp height. We place a cup (can be any object) within their grasp reach and ask them to gaze upon it. The system records the calculated 3D gaze point, and the real-time robot position, subtracting the two to find the offset in all three axis. This is then stored and used throughout trials for that user.

We have a fully functioning FSM that once activated can lead to continuous action implementations by the user without any interference by the system technicians. The grasp failure state allows for even failed tasks to simply be repeated until successful.

III. Experimental Evaluation

A. Evalutation of the 3D gaze calculation method

To evaluate the accuracy of the 3D gaze estimation method, we placed a target on the UR10 robot TCP, and programmed the robot to move to 10 points obtained randomly with uniform distribution, within the following range: $([0.25, 0.80], [0.15, 0.75], [0.35, 0.75])$. At each point, 50 gaze samples are obtained along with the real-time robot position. Gaze values are published at a frequency of $\approx 10Hz$, this is therefore equivalent to about 5 seconds at each point. The SMI eye-trackers require an initial period of random eye movements to obtain the pupil positions followed by a 3-point calibration, where the user fixates on three points with the system operator clicking on the screen at the point of fixation. The full experiment setup is shown in Figure 5. The 9-point board is placed at an $80cm$ distance from the user. Each subject is first asked to randomly fixate on the points for 1 minute. This is followed by the calibration step which is performed on the bottom left, top centre, and bottom right points on the board. The calibration is then

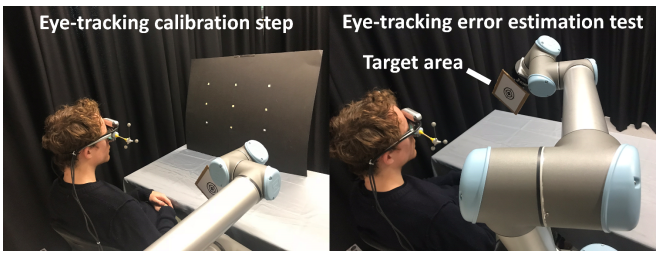

Fig. 5: The eye-tracking evaluation test setup.

verified and the board removed before the actual experiment starts. Errors in gaze estimation in x-axis, y-axis, z-axis and as Euclidean distance are measured and recorded. Results of the evaluation are provided and discussed in section IV.

B. Evaluation of the integrated robotic system

To evaluate the integrated system in an efficient manner, we selected the cup, bowl and table as objects of interaction. This is due to the cup having both "pick and place" and "pick and pour" functionalities within it, showing the grammar and context-based approach implemented in the form of the FSM. The system functions similarly with other objects discussed before, e.g. apples, oranges - these are shown in our video attachment.

A grid is drawn on the table in front of the user, creating 9 square boxes. The placement of the cup and bowl, and the target point on the table to drop objects on is randomised between these boxes for each trial. We chose to use the grid setup so that we would have a measure of success for object placement on the table, without indicating an exact point to our participants that they can fixate on, as doing so might help with the eye-tracking accuracy. The boxes are of an approximate size of $13cm \times 13cm$; but this is not an indication of the system resolution. That is instead, the outcome of the eye-tracking evaluation above and is reported in section IV. A schematic of the experiment setup can be seen in Figure 6.

The experiment tasks are: 1. Pick up the cup and place it back on the table at a different location and 2. Pick up the cup, pour it into the bowl, and then place it at a different location on the table. For pouring, small plastic balls are used to simulate a liquid, while conserving health and safety. Note that throughout the experiments, one of the researchers is constantly in possession of the UR10's emergency stop button, for added safety. Tasks are to be performed 5 times each. The users are asked not to contribute to the actuation and allow actions to be performed by the robotic system. As there is a learning curve involved with using the system, 3 attempts were allowed for each trial's first reach action; failures at later stages of a task are considered a task failure. Tasks are broken down into their lower-level actions, and the success/failure of these as well as the overall task success/failure are recorded as outcomes. Each participant completes a System Usability Scale (SUS) [27] subjective questionnaire after their test. These results are presented and discussed in section IV.

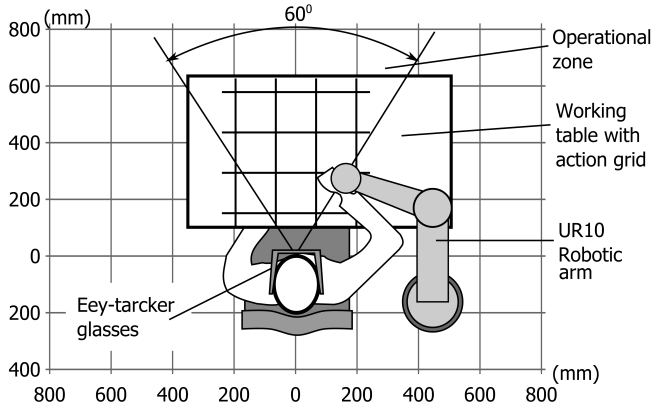

Fig. 6: The full integrated robotic system test setup.

IV. Results and Discussion

A. Gaze estimation results

For the evaluation of the 3D gaze estimation, 8 subjects were invited to our study, 25-30 in age, 6 male and 2 female. All subjects had normal or corrected to normal vision (wearing glasses which they were asked to remove). The first 10 gaze points from each trial are filtered out to avoid the transient effect of the user's gaze as the robot's move to a new point terminates. We use the 40 remaining gaze points for analysis, a total of 3280 gaze points. Each point has 3D coordinates of gaze location and ground truth (i.e. robot position). We average 3D coordinates of gaze data per trial to filter the gaze noise resulting in 80 data points. We use the Euclidean distance between the calculated gaze point and ground truth as the measure of accuracy.

In average, our system performs with the Euclidean error distance of $4.68 \pm 0.014cm$ (mean±SD). The Euclidean distance is normally distributed with 0.001 level (D'Agostino and Pearson's normality test: $p = 0.029$). To inspect the possible human factor influence, we perform one-way ANOVA. We found that the human factor does not affect Euclidean distance significantly with 0.001 level ($F(7, 72) = 2.345, p = 0.032$). Figure 7 shows the mean and standard deviation of the measure per subject. We can conclude that all subjects perform similarly.

We also checked the accuracy with respect to individual axes. We did not find significant correlation for X-axis (Spearman rank: $p = 0.085$), or the Y-axis errors (Spearman rank: $p = 0.715$), which correspond with the user's depth and horizontal axes directions respectively. The Z-axis or the height from the user's perspective, has significant correlation (Spearman rank: $correlation = -0.460; p << 0.01$). We believe this is due to the lower accuracy of pupil-tracking in extreme vertical positions. When eye balls go up they are less visible for the infrared cameras of the eye-tracker glasses. Overall, the results are consistent and show good performance of the system.

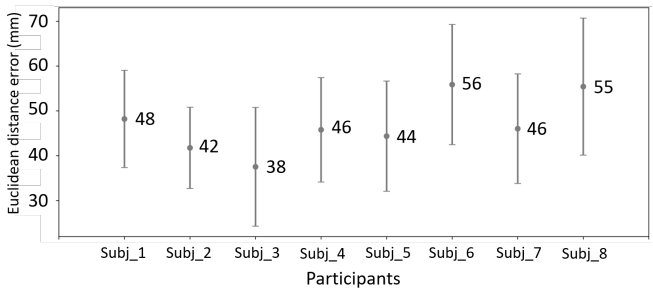

Fig. 7: The eye-tracking evaluation test results: Euclidean distance error in millimetres for all 8 participants as mean and standard deviation.

B. Full system results

For our full system tests, 5 participants joined the study. These were all male, 25-35 of age. Each experiment task consists of a number of lower-level actions. For task 1, pick and place on table, the actions are: Reach1, Grasp, Reach2 and Drop. For task 2, pick, pour in bowl, drop on table, the actions are: Reach1, Grasp, Reach2, Pour, Reach3, Drop.

Overall, for all participants and across all trials, task 1 was performed successfully in full 96% of the time. There were only two cases of failure: 1 instance of failing to grasp the cup and 1 instance of dropping the object slightly outside the indicated target box. Therefore, for task 1, across all trials and participants, Reach1 was successful 100% of the time, Grasp 96%, Reach2 100% and Drop 96% of the time. Task 2 was performed successfully in full 76% of the time. Failures were 3 instances of the final drop (was in the target area, but not placed upright); 2 instances of pour failures (pouring orientation change led to dropping the cup) and 1 instance of failure in the initial grasp of the cup. Therefore, for task 2, across all trials and participants, Reach1 was successful 100% of the time, Grasp 96%, Reach2 100%, Pour 91.7%, Reach3 100% and the final Drop, 87% of the time.

These results show mainly that the 3D gaze point estimation is well integrated with the system: all reaching cases are 100% successful, which is fully dependent on the 3D gaze point estimation being accurate. The Finite State Machine performed without errors throughout, making the implementation of these complex tasks possible with a very short training period (less than 5 minutes in all participants). Observed issues in the results are mainly within the pouring task, particularly at the pour action and its aftermath. The pouring orientation change had the effect of slightly moving the cup within the subjects' grasp (the cup and its contents are heavy), leading to either a premature drop of the cup, or a badly placed drop later on. This is mainly due to the design of the magnetic wrist attachment. We realised throughout the experiments that it does not provide the best support for the pouring action. This is an item that can be improved in the future.

All 5 participants filled in the System Usability Scale after their tests. Opinions on the system being "unnecessarily complex" are divided - 3 out of 5 choosing the borderline option, 1 agreed and 1 disagreed. On system "ease of use", 3 agree, 1 borderline and 1 disagree. On the system being "well integrated", 4 out of 5 agree, and 1 is borderline. On the system being "unpredictable" opinions are very divided: 1 agreed, 2 borderline, 1 disagreed and 1 strongly disagreed. On whether "most people would learn to use the system quickly", 4 out of 5 agreed (2 strong agreements) and 1 is borderline. On whether the system is "cumbersome to use", 3 disagreed and 2 are borderline. On whether they felt "confident using the system", 2 agree (1 strongly), 2 are borderline and 1 disagrees. On whether they "needed to learn a lot before they could use the system", all users disagree - 2 of them strongly.

These results generally show that the system was easy to learn for the users and not cumbersome. Most division of opinions are on whether the system is unpredictable and whether they felt confident using the system - though even in these cases results are favouring the system. We believe these two issues are related, however. As the users receive no direct feedback on how and when decisions and actions are made, the behaviour of the system might seem unpredictable, which will result in the users feeling less confident in its behaviour. This is an issue for us to look into as future work.

V. Conclusions

We presented a gaze-contingent robotic system for the restoration of reach and grasp capabilities. The main focus of our approach was to create a non-invasive, easy-to-learn and easy-to-use interface that would allow implementation of complex tasks made of sequences of several lower-level actions while conserving the simplicity of the interface. We follow the idea of action grammars: there are rules to our actions and how they can be combined together to create complex tasks. We implemented this, as a modelling of human cognitive behaviour, in the form of a finite state machine which monitors the human state, and implements actions on the robotic system when the necessary conditions are in place. This adds safety to the interaction, while making the implementation of complex tasks easy.

The user's gaze is monitored in high stability and accuracy through a new 3D gaze estimation method presented within this paper, fulfilled through integrating eye-tracker glasses and a depth camera. The objects within the user's environment are recognised using a convolutional neural network running on ego-centric camera images, allowing us to understand the context of the user's environment and interaction. Combining these, we are able to understand which object the user is looking at, whether they are interested in interacting with that object, and where that object is located in 3D space. Once these are fed to the finite state machine, it can direct the robotic system to implement complex sequences of actions for the user. Example tasks of "pick and place", as well as "pick, pour and place" were run with participants to test the system's performance and usabiltiy. Note that these are indeed, examples, and that actions and tasks can be expanded without issues. Results showed successful implementation of 100% of reaching actions, as well as 96% success in the "pick and place" task, and 76% in the "pick, pour and place" task. Issues in task completion were not, however, related to the performance of the 3D gaze estimation, object recognition or finite state machine modules; but rather due to physical and mechanical design choices within the system, such as the magnetic wrist attachment which is not the best support for pouring actions. Since then we have developed new attachments which require further motion planning for full implementation - an example of these is shown in our video attachment.

The users also filled in subjective questionnaires which showed they were content with ease of use and integration of the system, but were not all feeling confident with the system, with some feeling that it is unpredictable. We believe this is

due to lack of feedback to the users indicating an imminent action, which results in lack of explainability and therefore unpredictability. We will look into improved feedback to the user as future work. We are also interested in making further use of the RGBD camera, particularly to implement SLAM for better understanding of the environment (e.g. for obstacle avoidance when moving around the table), and better tracking of the user's head, to possibly remove the Optitrack system entirely making the system more mobile, e.g. through the use of a wheelchair mounted robotic arm for reaching support. These will be pursued as future work.

REFERENCES

[1] E. Rocon, J. Belda-Lois, A. Ruiz, M. Manto, J. C. Moreno, and J. L. Pons Rovira, "Design and validation of a rehabilitation robotic exoskeleton for tremor assessment and suppression," Institute of Electrical and Electronics Engineers, 2007.

[2] M. Bortole, A. Venkatakrishnan, F. Zhu, J. C. Moreno, G. E. Francisco, J. L. Pons, and J. L. Contreras-Vidal, "The h2 robotic exoskeleton for gait rehabilitation after stroke: early findings from a clinical study," *Journal of neuroengineering and rehabilitation*, vol. 12, no. 1, p. 54, 2015.

[3] C. Cipriani, F. Zaccone, S. Micera, and M. C. Carrozza, "On the shared control of an emg-controlled prosthetic hand: analysis of user–prosthesis interaction," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 170–184, 2008.

[4] R. J. Sanchez, J. Liu, S. Rao, P. Shah, R. Smith, T. Rahman, S. C. Cramer, J. E. Bobrow, and D. J. Reinkensmeyer, "Automating arm movement training following severe stroke: functional exercises with quantitative feedback in a gravity-reduced environment," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 14, no. 3, pp. 378–389, 2006.

[5] A. Cunningham, W. Keddy-Hector, U. Sinha, D. Whalen, D. Kruse, J. Braasch, and J. T. Wen, "Jamster: A mobile dual-arm assistive robot with jamboxx control," in *Automation Science and Engineering (CASE), 2014 IEEE International Conference on*, pp. 509–514, IEEE, 2014.

[6] K. Englehart and B. Hudgins, "A robust, real-time control scheme for multifunction myoelectric control," *IEEE transactions on biomedical engineering*, vol. 50, no. 7, pp. 848–854, 2003.

[7] A. B. Ajiboye, F. R. Willett, D. R. Young, W. D. Memberg, B. A. Murphy, J. P. Miller, B. L. Walter, J. A. Sweet, H. A. Hoyen, M. W. Keith, *et al.*, "Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: a proof-of-concept demonstration," *The Lancet*, vol. 389, no. 10081, pp. 1821–1830, 2017.

[8] K. Muelling, A. Venkatraman, J.-S. Valois, J. E. 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, no. 6, pp. 1401–1422, 2017.

[9] J. E. Downey, N. Schwed, S. M. Chase, A. B. Schwartz, and J. L. Collinger, "Intracortical recording stability in human brain–computer interface users," *Journal of neural engineering*, vol. 15, no. 4, p. 046016, 2018.

[10] W. W. Abbott and A. A. Faisal, "Ultra-low-cost 3d gaze estimation: an intuitive high information throughput compliment to direct brain–machine interfaces," *Journal of neural engineering*, vol. 9, no. 4, p. 046016, 2012.

[11] I. Ktena, W. Abbott, and A. A. Faisal, "A virtual reality platform for safe evaluation and training of natural gaze-based wheelchair driving," 2015.

[12] P. M. Tostado, W. W. Abbott, and A. A. Faisal, "3d gaze cursor: Continuous calibration and end-point grasp control of robotic actuators," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 3295–3300, IEEE, 2016.

[13] S. Dziemian, W. W. Abbott, and A. Aldo Faisal, "Gaze-based teleprosthetic enables intuitive continuous control of complex robot arm use: Writing & drawing," 2016.

[14] P. Majaranta and A. Bulling, "Eye tracking and eye-based human–computer interaction," in *Advances in physiological computing*, pp. 39–65, Springer, 2014.

[15] C. Breazeal and B. Scassellati, "A context-dependent attention system for a social robot," *rn*, vol. 255, p. 3, 1999.

[16] T. Yonezawa, H. Yamazoe, A. Utsumi, and S. Abe, "Gaze-communicative behavior of stuffed-toy robot with joint attention and eye contact based on ambient gaze-tracking," in *Proceedings of the 9th international conference on Multimodal interfaces*, pp. 140–145, ACM, 2007.

[17] G. Castellano, A. Pereira, I. Leite, A. Paiva, and P. W. McOwan, "Detecting user engagement with a robot companion using task and social interaction-based features," in *Proceedings of the 2009 international conference on Multimodal interfaces*, pp. 119–126, ACM, 2009.

[18] K. Fujii, G. Gras, A. Salerno, and G.-Z. Yang, "Gaze gesture based human robot interaction for laparoscopic surgery," *Medical image analysis*, vol. 44, pp. 196–214, 2018.

[19] H. He, Y. She, J. Xiahou, J. Yao, J. Li, Q. Hong, and Y. Ji, "Real-time eye-gaze based interaction for human intention prediction and emotion analysis," in *Proceedings of Computer Graphics International 2018*, pp. 185–194, ACM, 2018.

[20] Q. Li, C. Xiong, and K. Liu, "Eye gaze tracking based interaction method of an upper-limb exoskeletal rehabilitation robot," in *International Conference on Intelligent Robotics and Applications*, pp. 340–349, Springer, 2017.

[21] T. Carlson and Y. Demiris, "Using visual attention to evaluate collaborative control architectures for human robot interaction," in *Proceedings of New Frontiers in Human-Robot Interaction: A symposium at the AISB 2009 Convention*, no. CONF, SSAISB, 2009.

[22] G. Gautam, G. Sumanth, K. Karthikeyan, S. Sundar, and D. Venkataraman, "Eye movement based electronic wheel chair for physically challenged persons," *International journal of scientific & technology research*, vol. 3, no. 2, pp. 206–212, 2014.

[23] Q. X. Nguyen and S. Jo, "Electric wheelchair control using head pose free eye-gaze tracker," *Electronics Letters*, vol. 48, pp. 750–752, June 2012.

[24] T. F. Bastos-Filho, F. A. Cheein, S. M. T. Muller, W. C. Celeste, C. de la Cruz, D. C. Cavalieri, M. Sarcinelli-Filho, P. F. S. Amaral, E. Perez, C. M. Soria, *et al.*, "Towards a new modality-independent interface for a robotic wheelchair," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 3, pp. 567–584, 2014.

[25] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.

[26] C. Auepanwiriyakul, A. Harston, P. Orlov, A. Shafti, and A. A. Faisal, "Semantic fovea: real-time annotation of ego-centric videos with gaze context," in *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, p. 87, ACM, 2018.

[27] J. Brooke *et al.*, "Sus-a quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.