General Hand Guidance Framework using Microsoft HoloLens

David Puljiz[1], Erik Stöhr[1], Katharina S. Riesterer[1], Björn Hein[1,2], Torsten Kröger[1]

Abstract— Hand guidance emerged from the safety requirements for collaborative robots, namely possessing joint-torque sensors. Since then it has proven to be a powerful tool for easy trajectory programming, allowing lay-users to reprogram robots intuitively. Going beyond, a robot can learn tasks by user demonstrations through kinesthetic teaching, enabling robots to generalise tasks and further reducing the need for reprogramming. However, hand guidance is still mostly relegated to collaborative robots. Here we propose a method that does not require any sensors on the robot or in the robot cell, by using a Microsoft HoloLens augmented reality head mounted display. We reference the robot using a registration algorithm to match the robot model to the spatial mesh. The in-built hand tracking and localisation capabilities are then used to calculate the position of the hands relative to the robot. By decomposing the hand movements into orthogonal rotations and propagating it down through the kinematic chain, we achieve a generalised hand guidance without the need to build a dynamic model of the robot itself. We tested our approach on a commonly used industrial manipulator, the KUKA KR-5.

I. INTRODUCTION

With the emergence of collaborative robots [1], such as Universal Robotics' UR series, KUKA's LBR iiwa and others, hand guidance has become ubiquitous. This is due to the safety requirements needing joint-torque (JTS) or similar sensors on joints, but also allowed the gravity compensation mode (hand guidance), which allows easy teaching of new trajectories, even by lay-users. Such ease of teaching and reprogramming is of extreme importance to expand robotics to small and medium enterprises, as well as in flexible manufacturing paradigms [2]. Hand guidance on industrial robots in the mean time has been marginal, usually requiring external force-torque or other sensors in the robot cell. Traditional teaching pedants are still the most used tool to program industrial robots, despite requiring extensive training to use and being slow when it comes to reprogramming. Besides industrial settings, hand guiding is also important in medical robotics [3].

Beyond simply easing trajectory teaching, hand guidance is an essential part of Kinesthetic learning, a subfield of Imitation Learning [4]. The goal of imitation learning is to teach robot skills purely from human actions. This can be done either by observation and tracking of human movements and actions, adapting them to a robot, or by a human directly moving the robot, either real or virtual, to demonstrate how they would perform the action. The later is comprised
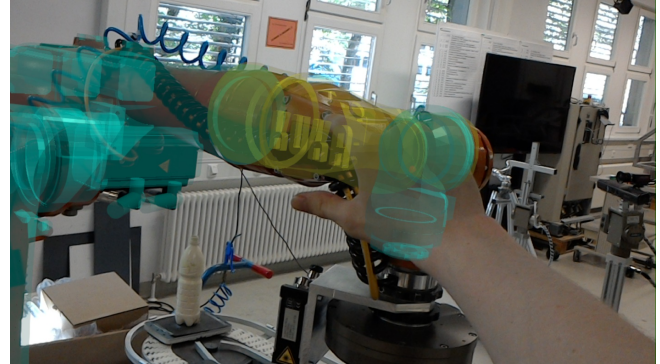

Fig. 1. The user moving their hand next to the robot link they want to move. Note that the link turns yellow to indicate which link will be moved

of kinesthetic teaching and teaching through teleoperation. Imitation learning can greatly reduce the search space in learning new tasks, as it already receives positive examples from demonstrations.

The use of Augmented Reality (AR) in the field of robotics is a long yet sporadic research field. Due to the lack of practicality and robustness in early head-mounted devices (HMDs), most of the earlier research used either camera and screen [5] or a projector [6] to display information. The release of the Microsoft HoloLens AR-device in 2016 marked the emergence of practical HMDs, allowing more flexible applications not bounded to an already set-up environment. Wearable AR has been used to ease robot programming [7], knowledge patching for imitation learning [8], task planning for collaborative human-robot workspaces [9] and for the display robot information to humans [10], among other things.

A. Related work

Methods of implementing hand guidance without JTSs already exist. Moe *et al.* use a Microsoft Kinect and a smartphone-based accelerometer to perform hand tracking and guide the end-effector of an industrial robot [11]. This approach, however, was limited to 5DOF. Furthermore by just driving the end-effector one cannot make use of extra degrees of freedom that redundant joints provide.

Lee *et al.* proposed a generalised method of hand guidance by torque control, based on the dynamic model of a robot, the motor current, and a joint friction model [12]. The approach requires experiments to determine the friction model of each joint, as well as possessing a dynamic model. Furthermore the robot is confined to using a torque controller. Finally the external force needed to move the end-effector was found

[1]Intelligent Process Automation and Robotics Lab, Karlsruhe Institute of Technology, Karlsruhe, Germany david.puljiz@kit.edu, ujeji@student.kit.edu, uxecu@student.kit.edu, bjoern.hein@kit.edu, torsten@kit.edu
[2]Karlsruhe University of Applied Sciences, Karlsruhe, Germany

to be 1.23-4.83 times greater than approaches based on JTS. Ideally the force would be close to zero, which isn't the case even for JTS methods.

A similar approach was taken by in [13], requiring a dynamic model and friction model identification, using a voting system and admittance control to move the robot. These two factors, as with the previous approach, prevent it to be a truly "plug-and-play" method.

In [14] a method without friction model identification was used with only a feed-forward term for partial friction compensation. The force required to move larger robots unfortunately became too large for human operators, due to uncompensated friction.

Teaching robot trajectories in virtual reality is also possible, however that requires either exact modelling of the workspace, or additional sensors to track it's status in real time, which we would like to avoid.

In this paper we propose an easily transportable method based on wearable sensors, thus not requiring any sensors on or around the robot. Our method requires only a kinematic model and works out of the box, not requiring any experiments or special setup before the use. Furthermore it can work with any controller, no force needs to be directly applied to the robot, and the sensitivity of the movement can be swiftly modified on-line. Finally it can be easily coupled with other wearable, AR-based robot programming and visualisation modalities. Particular interaction can be seen with [8] allowing both Kinesthetic teaching and knowledge patching on a single device.

B. Contributions

This paper presents three main contributions to the state of the art:

1) We develop a freely tuneable, flexible system for hand guidance, which does not need any sensors on or around the robot, any special identification experiments before use, or any specific controllers. It works with any type of robot, and, if the robot is inaccessible, it allows teleoperation.
2) We incorporate and test a semi-autonomous robot to headset referencing system that successfully overlays the virtual robot over the real one and tracks the position of the headset in reference to the robot.
3) We integrate these interaction modalities on a widely use AR device, using open-source tools. This allows the solution to be easily compatible with other AR interaction modalities and programs, thus increasing the scope of AR in human-robot interaction (HRI) systems

The structure of this paper is as follows. The next section presents the system overview and the referencing method we used to overlay the digital robot onto the physical one. In Section III we describe how we generate desired joint states from hand motions and the control architecture. Section IV describes the experiments we conducted and the results. Finally, conclusions are drawn and future work is outlined in Section V.

II. System and Referencing

The system consists of three main parts - the HoloLens, a computer and the KR-5 robot. The computer is running the open-source Robot Operating System (ROS) [15], and is connected to the KR5. The workflow is as follows - the AR device is started and connects to the computer running ROS, which sends it the Universal robot description file (urdf) and associated visual and collision meshes. This allows the device to generate a virtual robot on-line with the same geometry and kinematics as the real one.

The user can then choose to place the robot manually or to use the semi-automatic referencing. The manual mode likely will not result in a good match between the real robot and the "hologram", as is commonly referred in the HoloLens literature, and is intended for teleoperation scenarios where the robot is either inaccessible or when it's hard to manually reach particular links due to its size. In the latter case the hologram can also be resized.

The semi-automatic mode requires the user to place a "seed" hologram near the robot base and orient it approximately to the front of the robot. A registration algorithm is then used to refine the rough guess of the user, both increasing the precision and reducing the user strain of a purely manual method. The algorithm is described in detail in Section II-A.

Using the in-built hand-tracking and localisation capabilities of the HoloLens, we detect when the hand is near the robot by identifying if the hand is inside a resized convex collision mesh, which is generated from the meshes received from the computer. If the hand is inside this active zone, the motions of the hand are converted to the desired joint values of the robot. ROS Control [16] is then used to select the desired controller. These steps are described in Section III. Desired trajectories can be saved and replayed through a menu on the HoloLens itself. The UI can be seen in Fig. 2

A. Referencing Algorithm

The user first positions the seed hologram, in this case a cube, near the base of the robot and orients its z-axis(the axis is also visualised) to point approximately out of the front of the robot. We assume that the robot is static at the moment this is done and that the joint values are known to the controlling computer which in turn sends them to the AR device. The seed serves a dual purpose. Firstly it limits the search space of the registration algorithm, thus making convergence faster as well as preventing the registration to get stuck in a local minimum. Secondly it defines the centre of the area of the spatial mesh to be extracted. The spatial mesh is automatically generated by the HoloLens. We extract all the mesh in 2.5 meter radius of the seed hologram. The size can automatically be determined by the bounding box of the robot itself. The mesh is then sampled, converted to a point cloud and sent to the computer. Both the number of triangles per cubic meter and the sampling radius can be controlled resulting in larger or smaller point clouds. On the computer, the cloud is filtered using the Moving Least Squares Method and then either a standard Iterative Closest
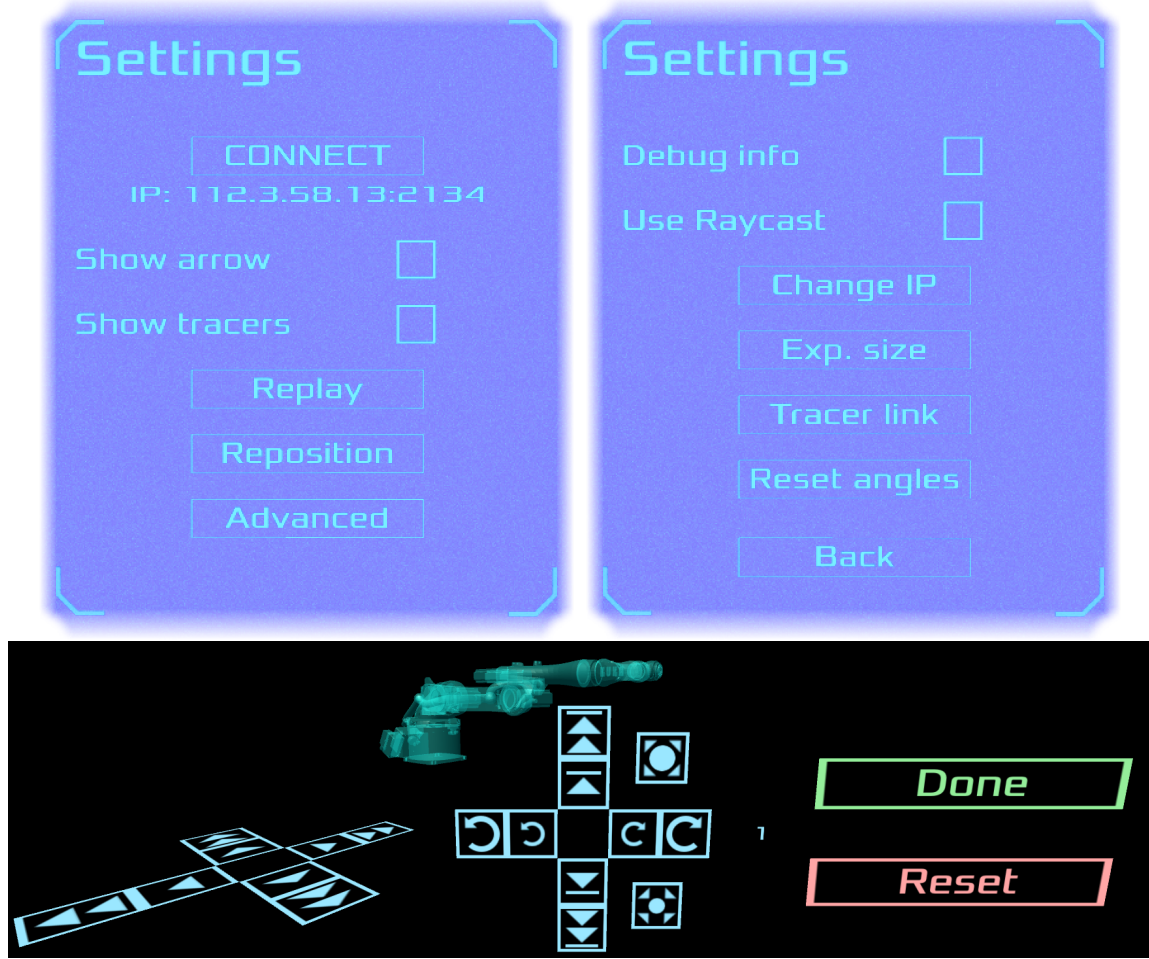
5186

Fig. 2. The UI of the application. On the top left one can see the main menu following the user. It allows to 1) open connections with a PC running the required ROS nodes 2) show arrows or tracers that illustrate better the hand movements detected 3) replay the trajectory performed 3)reposition and resize the robot (illustrated on the bottom picture) 4) open the advanced menu shown on the top left; The advanced menu is used for debugging, changing the IP and expanding the size of the space in which the hand movement activates robot movements; On the bottom options for repositioning and resizing the robot are shown.

Point (ICP) algorithm [17] or the Super4PCS [18] is used to refine the users initial guess by registering the model to the scene point cloud. This was implemented using the Point Cloud Library [19]. The evaluation of the approach is described in Section IV.

III. JOINT COMMAND CALCULATION AND CONTROL

Once a successful overlay is achieved, a resized convex meshes around the virtual (and now also real) robot links define the area in which hand movements translate to robot movements. The size of the area can be freely modified online through a holographic menu. A pre-programmed *Hold* gesture, defined in the HoloLens' specifications, is used to virtually push and pull the robot links.

When the user makes the *Hold* gesture inside the mesh of link j whose parent joint at position $s_{j,t}$ has a rotation axis $a_{j,t}$, the positions of the hand in the current frame h_t and the position of the hand in the previous frame h_{t-1} are projected unto the plane defined by $s_{j,t}$ and $a_{j,t}$ as per (1):

$$p_{j,t} = h_t - \frac{h_t \cdot a_{j,t}}{\|a_{j,t}\|^2} a_{j,t}, \quad p_{j,t-1} = h_{t-1} - \frac{h_{t-1} \cdot a_{j,t}}{\|a_{j,t}\|^2} a_{j,t} \tag{1}$$

The normalised vectors of the projection in relation to $s_{j,t}$ are:

$$v_{j,t} = \frac{p_{j,t} - s_{j,t}}{\|p_{j,t} - s_{j,t}\|} \quad , \quad v_{j,t-1} = \frac{p_{j,t-1} - s_{j,t}}{\|p_{j,t-1} - s_{j,t}\|} \tag{2}$$

The angle between the two vectors, and therefore the change in joint angle is

$$\Delta\theta_{j,t} = \arccos\left(v_{j,t-1} \cdot v_{j,t}\right) \cdot \text{sign}(a_{j,t} \cdot (v_{j,t-1} \times v_{j,t})) \tag{3}$$

The new desired joint angle is therefore

$$\theta_{j,t} = \begin{cases} \theta_{j,t-1} + K\Delta\theta_{j,t} & \theta_{j,min} \le \theta_{j,t} \le \theta_{j,max} \\ \theta_{j,t-1} & \text{otherwise} \end{cases} \tag{4}$$

Where K is a tunable motion scaling factor.

5187

This angle update however covers only part of the hand motion, to get the full motion we need to propagate the remaining motion down through the kinematic chain. We rotate h_{t-1} around $a_{j,t}$ centred in $s_{j,t}$ by the angle $(\theta_{j,t} - \theta_{j,t-1})$

$$r_{j,t-1} = q_{j,t} \cdot (h_{t-1} - s_{j,t}) \cdot q_{j,t}^{-1} + s_{j,t} \tag{5}$$

Where $q_{j,t}$ is the quaternion representing the rotation around $a_{j,t}$, $q_{j,t}^{-1}$ is it's conjugate, and the $\cdot$ represents the Hamiltonian product. The quaternion $q_{j,t}$ is defined as

$$\hat{q_{j,t}} = (\cos\left(\frac{\theta_{j,t} - \theta_{j,t-1}}{2}\right), \sin\left(\frac{\theta_{j,t} - \theta_{j,t-1}}{2}\right)a_{j,t}),$$
$$q_{j,t} = \frac{\hat{q_{j,t}}}{\|\hat{q_{j,t}}\|} \tag{6}$$

The same computation is then reiterated for each joint down the kinematic chain until $r_{j,t-1} = h_t$ or there are no more joints left. In the latter case the complete motion of the hand is not reproduced by the robot due to kinematic constrains. In Fig. 3 one can see a graphical representation of the equations.
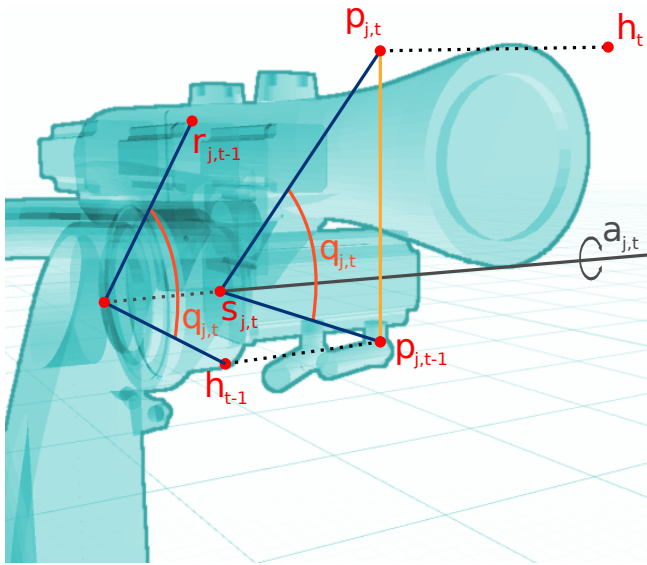

Fig. 3. Graphical representation of (1) - (6). The hand motion between the two consecutive frames has been vastly exaggerated for the sake of clarity

The vector of joint position updates is then sent to ROS where any controller can be selected through ROS Control. If the controller in question is Cartesian, the desired joint values can be converted to the desired end-effector pose via forward kinematics.

Given that in the majority of cases dragging the end-effector may be preferable, a holographic sphere is placed around the end-effector which can be dragged and rotated around the three axis of the world coordinate system. The pose of the sphere then indicates the desired pose of the end effector. For control in joint-space the inverse kinematic solution is used that minimizes the total change of all the joint angles. In our case we use the MoveIt! framework to perform both forward and inverse kinematics as needed.
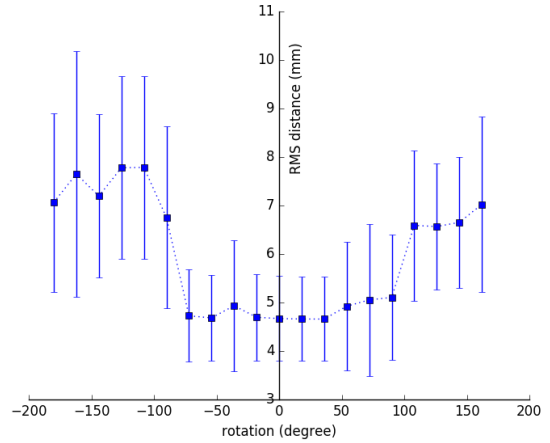

Fig. 4. The influence of rotation on our registration algorithm. One can see that the algorithm is robust to rotational errors of the seed hologram, meaning that the user guess does not have to be overly precise. One can also see that that error is not mirror due to the robot being asymmetric

IV. EXPERIMENTS AND RESULTS

To prove the validity of our approach we made two sets of experiments. In the first set, we tested how well does our referencing algorithm perform. This is crucial in the cases of proximal HRI where the robot need to be guided as similarly as possible to JTS-based methods. In the second set of experiments we test how well our control strategy translates hand motion to robot motion.

In regards to referencing, two main questions needed to be answered. First, how does it compare to purely manual referencing. Secondly, how precise does the positioning of the seed hologram influence the referencing error.

The input point clouds for the registration algorithm were divided into two options, namely a big one (1,240,000 triangles per cubic meter and 256,000 samples per mesh) and a small one (1,000 triangles per cubic meter and 16,000 samples per mesh), with the conversion taking 1 min, and 3 seconds respectively.

TABLE I

THE MEAN, MINIMUM, MAXIMUM AND STANDARD DEVIATION IN MILLIMETRES OF THE CONDUCTED TESTS. THE FIRST ROW REPRESENTS THE 12 ORIGINAL HUMAN GUESSES. THE ICP-ROTATION AND ICP-TRANSLATION ARE THE TESTS CONDUCTED BY ROTATING AND TRANSLATING THE ORIGINAL USER GUESSES RESPECTIVELY

	mean(mm)	min(mm)	max(mm)	σ (mm)
User guess	27.53	3.65	138.87	44.52
ICP - all	4.91	3.22	14.42	1.39
ICP - rotation	5.90	3.22	14.21	1.90
ICP - translation	4.74	3.22	14.41	1.20

Preliminary experiments showed that in our test scenario Super4PCS had poor performance. The reason for this being that it requires segmented out models from the scene to work, being a global descriptor algorithm. Our test cell has a table very close to the robot, as seen in Fig. 1 which was not possible to be segmented-out fully. Furthermore in
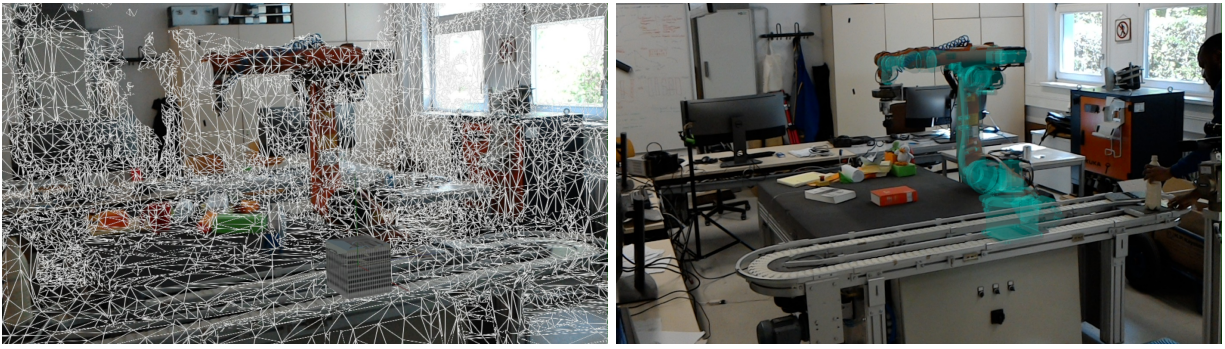
Fig. 5. The results of the referencing algorithm; (left) The HoloLens spatial mesh and the seed hologram positioned by the user; (right) The hologram overlayed on the real robot

experiments with a similar model, the KR-6, it showed very wide precision deviation based on the parameters chosen. ICP on the other hand proved much more robust in case of parameter changes. The small cloud was chosen as it had comparable performance with much less processing time.

The root mean square (RMS) distance between the closest points in the two point clouds was used as a metric for the precision of the referencing. As a second evaluation, the matches were visually inspected, as some bad matches could have quite low RMS distances if close to other objects in the scene. These results were disregarded. Twelve user guesses and point clouds were taken. Each user guess was then rotated in steps of 18° to test the influence of imprecise rotation of the seed hologram. The influence of rotation on the ICP can be seen in Fig. 4. Keeping the original rotation, the seed algorithm was translated in a 1 m volume around the initial guess with a step of 0.1 m. In Table I one can see the precision of original user guesses, the ICP subject to rotation of the seed algorithms, the ICP subject to translation, and the average statistics of all cases. One can see that the ICP performs much better and with less deviation than a purely manual method. The referencing result can be seen in Fig. 5.

Tests of the the control strategy were performed using $K = 1$ and a Reflexxes interpolated joint position controller [20]. Moving each link, we compared the positions of the end-effector to the hand position, as well as tracked the desired joint state provided by our algorithm, the control signal of the Reflexxes controller, and the actual joint state of the robot. The robot used was an industrial KUKA KR-5 ARC manipulator with no JTS. In Fig. 6 one can see the example where link 3 is moved. The end-effector, and therefore link 3, motions follow closely the hand motions. One can also see that, although the hand commands suffer from shaking as they were unfiltered, the inertia of the robot filters out this signal and the joint states are smoother than the commands. That said a tunable band-stop filter, or signal smoothing for small, shaky hand motions could be a helpful addition.

Also visible in Fig. 6, the desired joint values follow closely what one would intuitively expect from our command strategy, with the base joint (joint 1) and elbow joint (joint3)

receiving the majority of move commands. The non-zero move command from the shoulder joint (joint 2) is due to the fact that when $s_{3,t}$, h_t and h_{t-1} are collinear, the projection angle is zero, and therefore the next joint, joint 2, needs to perform the movement.

V. CONCLUSIONS AND FUTURE WORK

To extended the option of using hand guidance to any robot possessing an urdf description and a 3D model, we developed a system that uses ICP-based registration to find a robust transform between the Microsoft HoloLens headset and the robot. After that the user has the option to perform either telemanipulation or direct hand guidance near the robot, either through end-effector dragging, or through pushing and pulling specific links which is very useful for robots with redundant kinematics. We proved the feasibility of this approach by testing both the robustness of the referencing strategy and the control strategy using a KR-5 industrial manipulator. The referencing performed better than a pure human guess, with good resilience to errors in translation and rotation of the seed hologram. Meanwhile the control strategy successfully moved the robot in a way that closely follows the motions of the user's hand.

The in-built hand-tracking used in this paper had difficulties tracking the hand in front of black backgrounds, likely due to the absorption properties, and when in close proximity to objects, due to the fact that the hand cannot be segmented out from the background. To alleviate these problems, future implementations will transition to the method proposed in [21], which can deal both with contact cases and a dark background as it uses RGB images instead of an infrared depth stream. Furthermore it can track the hand of the user in any pose, meaning that the user's arm does not need to adhere to the few pre-programmed gestures of the in-built hand tracking.

Future work also includes tests with more robots to detect any failure cases of either referencing or the control strategy, and measuring the referencing error with respect to the ground truth. Adapting the speed and timing of the learned trajectory should also be addressed. Finally a user study should be carried out where our approach is compared with one based on JTSs to evaluate the general usability, as well
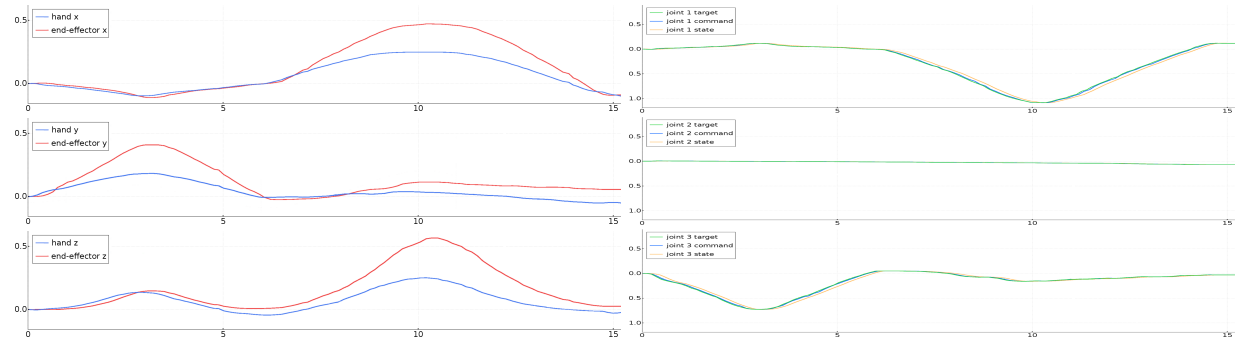
Fig. 6. The Cartesian movements of the end-effector and joints 1-3 while moving the third link for 15 seconds. (left) The movements of the hand strongly correlate with those of the end-effector and therefore link 3, as the three joints up the kinematic chain weren't changed. The movements of the end-effector are larger than that of the hand as the end-effector is further away from the third joint. The y axis is in meters and the x axis in seconds (right) The joint commands of joints 1-3, with joint 3 being the closest to link 3. As the axis of rotations of joints two and three are parallel, one can see almost no change in joint two. One can also see the state of the robot lags behind the commands due to inertia. The y axis is in radians and the x in seconds

as determining how visualising the robot hologram and hand positions under referencing errors might impact user comfort.

ACKNOWLEDGMENT

REFERENCES

[1] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schäffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald *et al.*, "The kuka-dlr lightweight robot arm-a new reference platform for robotics research and manufacturing," in *Robotics (ISR), 2010 41st international symposium on and 2010 6th German conference on robotics (ROBOTIK)*. VDE, 2010, pp. 1–8.

[2] R. D. Schraft and C. Meyer, "The need for an intuitive teaching method for small and medium enterprises," *VDI BERICHTE*, vol. 1956, p. 95, 2006.

[3] M. Hanses, R. Behrens, and N. Elkmann, "Hand-guiding robots along predefined geometric paths under hard joint constraints," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sept 2016, pp. 1–5.

[4] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[5] R. Bischoff and A. Kazi, "Perspectives on augmented reality based human-robot interaction with industrial robots," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 4. IEEE, 2004, pp. 3226–3231.

[6] A. Gaschler, M. Springer, M. Rickert, and A. Knoll, "Intuitive robot tasks with augmented reality and virtual obstacles," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6026–6031.

[7] C. P. Quintero, S. Li, M. K. Pan, W. P. Chan, H. F. M. V. der Loos, and E. Croft, "Robot programming through augmented trajectories in augmented reality," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 1838–1844.

[8] H. Liu, Y. Zhang, W. Si, X. Xie, Y. Zhu, and S.-C. Zhu, "Interactive robot knowledge patching using augmented reality," in *2018 IEEE 21st International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1947 –1954.

[9] T. Chakraborti, S. Sreedharan, A. Kulkarni, and S. Kambhampati, "Alternative modes of interaction in proximal human-in-the-loop operation of robots," *CoRR*, vol. abs/1703.08930, 2017. [Online]. Available: http://arxiv.org/abs/1703.08930

[10] M. Walker, H. Hedayati, J. Lee, and D. Szafir, "Communicating robot motion intent with augmented reality," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '18. New York, NY, USA: ACM, 2018, pp. 316–324. [Online]. Available: http://doi.acm.org/10.1145/3171221.3171253

[11] S. Moe and I. Schjlberg, "Real-time hand guiding of industrial manipulator in 5 dof using microsoft kinect and accelerometer," in *2013 IEEE RO-MAN*, Aug 2013, pp. 644–649.

[12] S. D. Lee, K. H. Ahn, and J. B. Song, "Torque control based sensorless hand guiding for direct robot teaching," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 745–750.

[13] M. Ragaglia, A. M. Zanchettin, L. Bascetta, and P. Rocco, "Accurate sensorless lead-through programming for lightweight robots in structured environments," *Robotics and Computer-Integrated Manufacturing*, vol. 39, pp. 9 – 21, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S073658451500126X

[14] A. Stolt, F. B. Carlson, M. M. G. Ardakani, I. Lundberg, A. Robertsson, and R. Johansson, "Sensorless friction-compensated passive lead-through programming for industrial robots," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 3530–3537.

[15] 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. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, May 2009.

[16] S. Chitta, E. Marder-Eppstein, W. Meeussen, V. Pradeep, A. Rodríguez Tsouroukdissian, J. Bohren, D. Coleman, B. Magyar, G. Raiola, M. Lüdtke, and E. Fernández Perdomo, "ros_control: A generic and simple control framework for ros," *The Journal of Open Source Software*, 2017. [Online]. Available: http://www.theoj.org/joss-papers/joss.00456/10.21105.joss.00456.pdf

[17] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, Feb 1992.

[18] N. Mellado, D. Aiger, and N. J. Mitra, "Super 4pcs fast global pointcloud registration via smart indexing," *Computer Graphics Forum*, vol. 33, no. 5, pp. 205–215, 2014. [Online]. Available: http://dx.doi.org/10.1111/cgf.12446

[19] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[20] T. Krger, "Opening the door to new sensor-based robot applicationsthe reflexxes motion libraries," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 1–4.

[21] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt, "Ganerated hands for real-time 3d hand tracking from monocular rgb," in *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, June 2018. [Online]. Available: https://handtracker.mpi-inf.mpg.de/projects/GANeratedHands/