Augmented Reality Based Visual Force Feedback For Physical Human-Robot Interaction

Manfred Schönheits
German Aerospace Center (DLR)
Center for Lightweight Production Technology (ZLP)
86159 Augsburg, Germany
manfred.schoenheits@dlr.de

Abstract— **Physical human-robot interaction usually requires physical contact between humans and manipulators in any form. That involves forces and torques exerted on the manipulator as well as the human vice-versa. In contrast to motion and proximity which can be observed visibly, forces and torques cannot. A lot of work covers haptic force feedback interfaces or virtual environments for this purpose. In this paper, we introduce an approach to enable visual force feedback in the real world through augmented reality. We develop and test such a system using the Microsoft Hololens and discuss our results and potential future work and applications.**

I. INTRODUCTION

For both robot-based automation as well as human-robot interaction, amongst others, two physical quantities play a key role: force and torque. For instance, every payload attached to a manipulator generates a gravitational force by its mass and corresponding torques exerted on the joints of the robot. Physical human-robot interaction involves contact forces and torques e.g. when gripping, touching or guiding a manipulator. Other than factors like motion or proximity, these physical quantities are not visible to the naked eye.

In [1], visualization techniques for forces and torques in the context of robotics have been investigated using conventional display technologies. Visual force feedback in a virtual assembly environment has been evaluated in [2]. Teleoperation in an on-orbit scenario with haptic feedback is discussed in [3].

In this paper, we introduce a prototypical visual force feedback system based on augmented reality (AR). We make the assumption that such a system can be a beneficial contribution to pHRI systems, be it directly integrated into the interaction or indirectly during the design phase of such systems.

II. DESIGN CONSIDERATIONS

In this section, we outline the considerations that have been made for the design of the system. The idea was to make an operator interact with a manipulator or another person involved to be able to "see" forces and torques. The basic concept was to create digital representations of theses quantities and display them to the operator through an AR system at the world coordinates where they are applied. A few necessary guidelines had been identified as follows:

- The visual representation needs to have some functionality to reflect the proportional character of vector quantities in this context.
- The AR system needs to be calibrated to the device, i.e. the world transformation from the robot to the AR system must be determined.
- The system needs to have an interface to be fed with the signal from the source(s) in soft real-time.

III. VISUAL FORCE FEEDBACK SYSTEM IMPLEMENTATION

Currently, multiple AR systems are available on the market. Basically one can distinguish between handheld systems, e.g. Apple$^{\mathrm{TM}}$iPad with ARKit and wearables like the Magic Leap$^{\mathrm{TM}}$One or the Microsoft$^{\mathrm{TM}}$Hololens. One benefit of wearables is that they allow hands-free operation, this way enabling the operator to interact with a manipulator with the same freedom of motion and dexterity as without an AR system. For this reason and because it was already available, we used the Hololens for the matter of this paper.

A. Hololens Platform Setup

Applications can be developed for the Hololens in the form of UWP[1]-Apps. Rendering can be done using the DirectX-Graphics API directly, while it is common practice to use an engine like e.g. Unity$^{\mathrm{TM}}$because it allows to exploit functionality like the content pipeline and scripting infrastructure.

As a first step, a model of the robot was created in Unity, shown in figure 1. In this example, we are using a KUKA LBR iiwa 820 because it is the manipulator we have done our first tests with as it has reliable force and torque sensing built-in. Kinematic functions have been added to the model so that we are able to make the virtual robot follow the real robot's state.

For the visual representation of forces and torques, a model has been created in Blender$^{\mathrm{TM}}$, as shown in figure 2. This model allows to display forces and torques as arrows that are pointing in or wrapping around the corresponding directions respectively. The magnitude is represented by a scaling function, that scales the force and torque components corresponding to the input signal. For visualizing forces and

---

[1] Universal Windows Platform

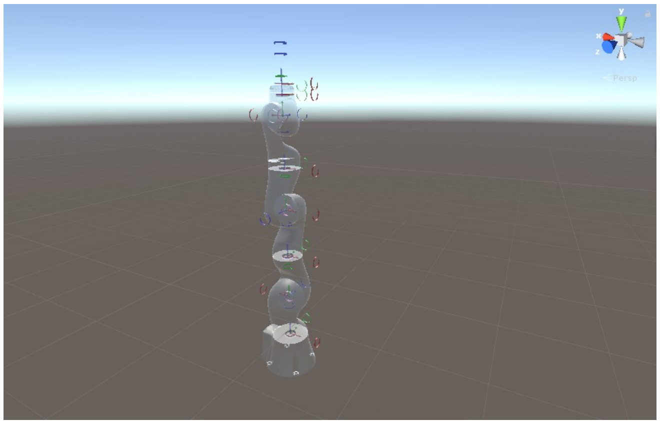

Fig. 1. LBR iiwa modeled in Unity

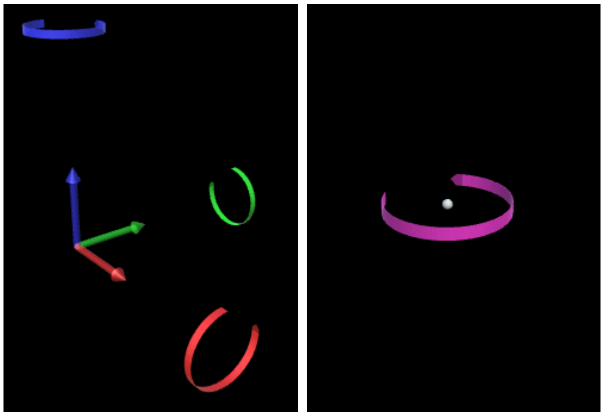

Fig. 2. Visual representations, component-wise (left) and joint-wise

torques resulting directly from a physical interface like e.g. a handle or a tool, a component-wise model was used. For projecting the torques exerted at the robot joints, a simpler joint-and-axis model was derived.

B. *Calibration*

In order to display the visualization at the right place in the real world, the digital model needs to be calibrated with respect to the real robot. For the following tests, two different calibration methods have been used:

Manual Interactive Calibration: The operator uses hand tracking capabilities to manipulate the digital model until it aligns with the real one.

Automated Marker-Based Calibration: A fiducial marker with a known offset is placed on or near the robot. A marker detection running on the HoloLens detects the marker pose. From those two transformations, the world calibration can be obtained.

Figure 3 shows not-yet calibrated scene as the application is loaded and the model after calibration. On the left, the virtual model displayed as a semi-transparent silhouette is offset to the left of the real robot. On the right, the virtual model and the real robot are aligned. Both methods have been found to provide results that are sufficient for this purpose, however they depend on the context like the experience of the user or the nature of the surroundings. Although it is an important thing we put effort in currently, an in-depth discussion of this topic is beyond the scope of this paper.

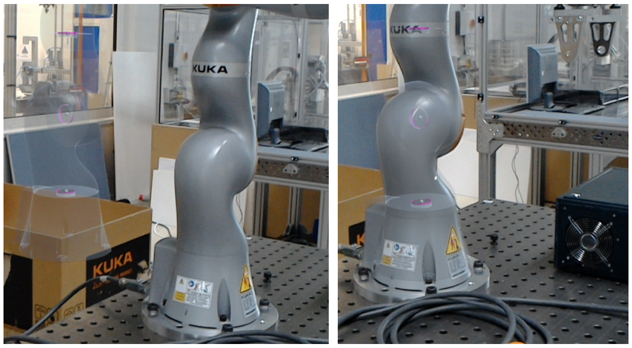

Fig. 3. Wrong initial world position of model (left) and correct alignment (right)

C. *Interfaces*

In our example, the LBR iiwa is the only signal source. The KUKA Sunrise Controller the iiwa is running on provides a Java API that allows reading out the torques exerted on the robot joints as well as the forces and torques applied to the flange of the robot or a tool mounted to it.

A simple UDP protocol was defined to establish the communication between the HoloLens and the iiwa. Clients for that protocol have been implemented in Java on the Sunrise controller and in C# for UWP on the Hololens. A schematic of the network setup is shown in figure 4.

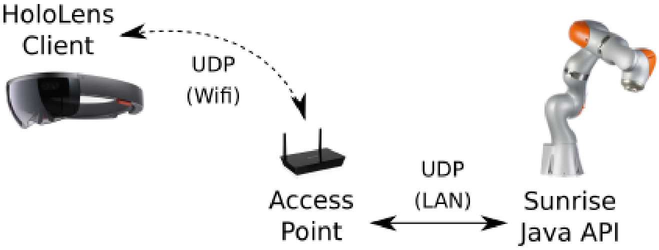

Fig. 4. Network setup

The data transmission rate was hard-coded and arbitrarily set to run at a 20Hz ($\equiv$ 50ms) cycle.

IV. TESTING

Once the setup was completed, first tests have been done. By pushing against an end-effector stub or the flange itself, forces were applied to the robot. Figures 5 and 6 show the operator's perspective while pushing in Y-direction and X-direction of the flange respectively. The sizes of the arrows pointing in the direction of the flange coordinate system axes correspond to the force applied in this direction.

One observation was that it is hard to manually apply force in precisely one direction only. Also, most of the time torques got exerted as well although not intended because usually there are levers from where the force is applied and where the force gets measured. Basically the system was working

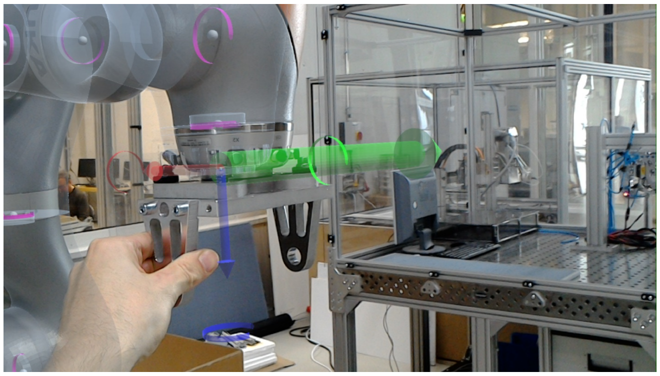

Fig. 5. Force applied in Y-direction of the robot flange

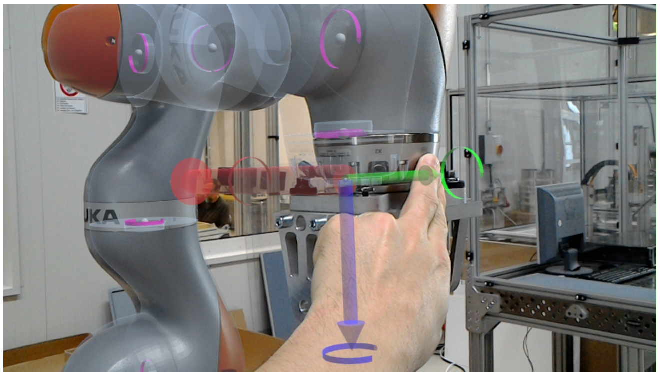

Fig. 6. Force applied in X-direction of the robot flange

as expected. Some fine-tuning had to be done to get the best usability, i.e. adjusting the minimum and maximum values for the scaling function to the actual measured forces. In this simple example already, the advantage of such a system compared to e.g. only having 2D-plots offline on your screen becomes apparent.

In a following step, the iiwa was integrated in one of the robot cells at the DLR site in Augsburg. Here, several ceiling mounted industrial robots are available. The iiwa was attached to one of them, resulting in an upside-down-configuration. In figure 7, one can see the visualization of the torques needed to compensate gravity and follow the programmed trajectory the robot is moving along by itself (picture was taken while the robot was moving). The pink arrows wrapping around the joint axes in a circular way correspond to the torque applied to each joint. The higher the torque exerted on a joint, the larger the diameter of the circle gets. The sign of the torque is represented by the circle going clockwise or counter-clockwise respectively.

Next, the iiwa was put into hand-guiding mode where the iiwa internally does gravity compensation yet follows the external torques applied to its joints. At this point, the iiwa had a small 3D-printed end-effector attached to it, which has a little handle that can be used for manipulation. A snapshot of the results can be seen in figure 8. At the moment the picture was taken, the operator was pulling the robot towards himself. One can see that the displayed vectors correspond to the force induced by the operator.

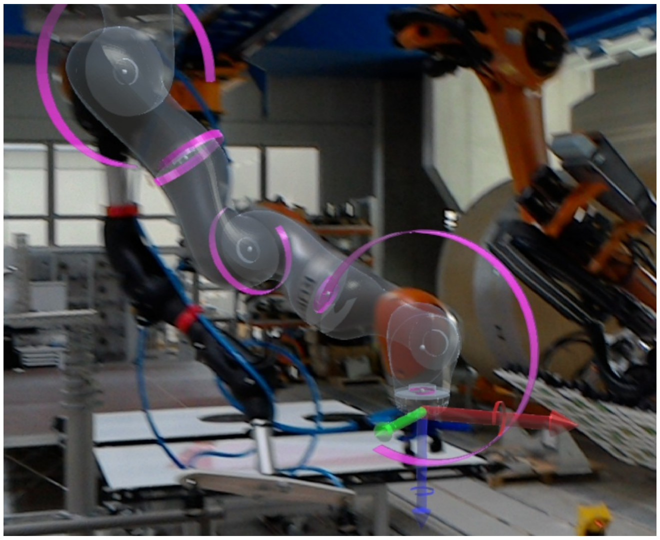

Fig. 7. Torques exerted at the robot joints

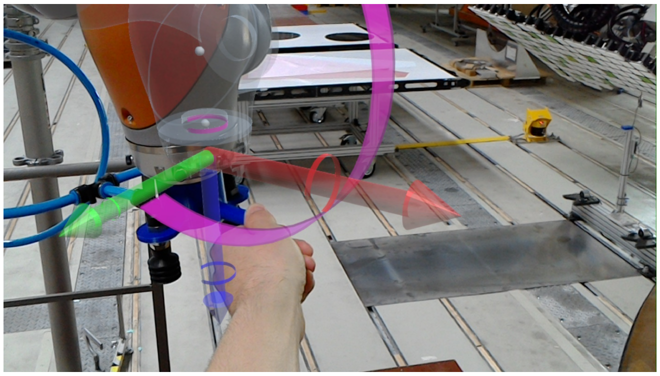

Fig. 8. Force- and torque-visualization when handguiding the iiwa

V. CONCLUSIONS

The visual force feedback system we have developed was found to produce useful and reasonable results for our experiments. However, it has been tested with only one device yet, the KUKA LBR iiwa. In the meantime a client for another platform, the Universal Robots UR10, has been implemented, but the work could not have been completed to make it into this paper.

Although some results seem to be promising, as the immediate visualization revealed some counter-intuitive effects, the benefit of the system can be seen questionable in our simple scenario. For instance, when hand-guiding the robot in a simple way like we did, the operator ”feels” the forces and torques as they generated by the operator as the input to the robot.

However, in a scenario that is more complex than ours, the advantage of a system like we have designed may emerge considerably. For example, when an operator is hand-guiding the robot, but the handle is mounted to the robot in a different place compared to where the robot touches an object or even another person.

Also such a system might be useful in scenarios like

[4] where the input impedance is different from the output impedance because of a mechanism that connects both parts of the system.

Another potentially good application could be mechanisms that are more complex than an articulated robot arm. Here, an AR-based visual force feedback system maybe could help debugging or during testing and design of the mechanical system.

Future work could include adapting the system to or using it in conjunction with different mechanisms as mentioned above. Furthermore, external force or torque-driven components like springs, extrinsic sensors and dampers could be included.

REFERENCES

[1] T. Hulin, K. Hertkorn, and C. Preusche, “Interactive features for robot viewers,” in *International Conference on Intelligent Robotics and Applications*, ser. Lecture Notes in Computer Science, C.-Y. Su, S. Rakheja, and H. Liu, Eds., vol. 7508. Springer, Oktober 2012, pp. 181–193. [Online]. Available: https://elib.dlr.de/76331/

[2] M. Sagardia, B. Weber, T. Hulin, C. Preusche, and G. Hirzinger, “Evaluation of visual and force feedback in virtual assembly verifications,” in *IEEE VR 2012*, März 2012. [Online]. Available: https://elib.dlr.de/75231/

[3] M. Sagardia, K. Hertkorn, T. Hulin, S. Schätzle, R. Wolff, J. Hummel, J. Dodiya, and A. Gerndt, “Vr-oos: The dlr’s virtual reality simulator for telerobotic on-orbit servicing with haptic feedback,” in *IEEE Aerospace*, März 2015. [Online]. Available: https://elib.dlr.de/102000/

[4] N. Badeau, C. Gosselin, S. Foucault, T. Lalibert, and M. E. Abdallah, “Intuitive physical human-robot interaction: Using a passive parallel mechanism,” *IEEE Robotics Automation Magazine*, vol. 25, no. 2, pp. 28–38, June 2018.