An Investigation into the Feasibility of Remote Object Manipulation Using Consumer Virtual Reality Hardware and Decoupled Mapping Control

Habib Aghasafari
Robotics Institute, Faculty of Engineering & IT
University of Technology Sydney, NSW, Australia
Habib.aghasafari@student.uts.edu.au

Abstract

With the occurrence of the COVID-19 pandemic, many health workers in hospitals, nursing homes and quarantine facilities were put at increased health risk in their workplace.
One effective solution to reduce this risk is to use Telerobotics which enables health workers to carry out their tasks remotely. Such a system must be able to perform a wide range of tasks in an unpredictable environment.
This research paper will focus on the pick and place of small objects, which is one of the most common tasks in health care settings.
This research designs a prototype telerobotic system using an innovative and new control method to remotely pick and place small objects and will perform the following test:
An operator will use a virtual reality headset to remotely control a robot arm located more than 44 kilometres away from the operator and pick several common everyday objects and place them into a container.
By developing this proof of concept prototype, this research will help to accelerate the adoption of telerobotic technologies in health care and other industries.

1 Introduction

Telerobotics is one of the most fascinating fields of robotics. It combines many areas of engineering. It allows a person to virtually travel to distant places and to do useful work in remote places that are dangerous to humans.

Many researchers and engineers have developed telerobotic systems in the past, but many were limited by the technology of their time.
Technological progress in recent decades has made it possible to significantly improve telerobotic systems.

Firstly, internet bandwidth and latency have significantly improved in the last two decades. We went from dial-up internet to ADSL, and now we are in the age of fibre optics and 5G internet technology.
Secondly, the input devices for tracking the position and orientation of hand movements in 3D space were not readily available.
Some researchers used another robot (usually at a smaller scale) to act as the master robot. Others used exoskeletons that could attach to human arms and hands [Klamt et al. 2019].
The main problem with these solutions is that they are very expensive and therefore out of everyday people's reach.
In recent years, this problem has been solved by the rise of consumer Virtual Reality (VR) hardware such as Oculus Quest, an all-in-one VR headset and controllers that can provide exceptional hand tracking and VR experience for just 299 US dollars. No other hardware or computer is needed [Oculus n.d.].
This paper will introduce an innovative, simple, yet highly effective way of mapping between a UR3 robot arm and a VR hand controller. This mapping will be used to develop an intuitive telerobotic control system that is highly effective in performing remote pick and place of small objects.

2 Related work

In 2017, a team of researchers from MIT had successfully developed a telerobotic system leveraging consumer virtual reality headsets and a Baxter robot. [Lipton et al. 2017]. They have published a video [MITCSAIL 2017] where they demonstrate their system's capabilities. This video has sped up, and the user and the robot are on the same local network.
The operator controls the end effector of the arms by manipulating a set of virtual blue spheres inside of the VR environment. This method is therefore not as intuitive as if the user could control the robot freely without the use of any virtual object.
Once they tried their system over a different network within the same building, they had to reduce their camera feeds to 5Hz. This shows that their software or network solution for communicating was inefficient. For

comparison, this prototype uses two camera feeds with similar resolution and achieved 30Hz for both feeds and 50Hz for tracking data feed over a 44km distance.
Another team from the University of Washington have developed a telerobotic system for underwater remediation of military munitions [Gharaybeh et al. 2019]. They also use a VR headset and hand controllers together with underwater LIDAR and cameras to build a 3D simulation environment of the working site.
The problem is that this LIDAR scan takes about a minute to complete. The operator can then control the robot arm within this 3D environment using direct inverse kinematics which has a slow rate of 19Hz. By comparison, the inverse kinematics designed for this prototype has a calculation rate of around 3400Hz.
This system is therefore highly inefficient. It cannot react to dynamic changes in the robot's environment.
A team of researchers from Mexico has developed a prototype of an areal drone with an attached manipulator controlled with a VR headset and hand controllers [Verd´ın et al. 2021].
A team of researchers from Italy and New Zealand has developed a prototype telerobotic system with a UR5 robot arm and VR headset and controllers [De Pace et al. 2021]. This system works well, but the experiment has only been conducted over a local network.
[Van de Merwe et al. 2019] made a simulation only telerobotic system using VR. The usefulness of VR in Telerobotics was also highlighted by [Franzluebbers & Johnsen 2019].
In recent years, the ANA AVATAR XPRIZE has accelerated research and development in the Telerobotics field. This is a four-year award that started in 2018 and will give 10 million US dollars to the best teams that develop telerobotic systems "that can transport human presence to a remote location in real-time" [Xprize 2021]. The next two teams are both participants in this competition.
Unfortunately, as of the writing of this paper, they have not published any research papers about their projects which means only limited information about their work is available. Still, it is useful to review their work and to understand what is achievable with the cutting edge of telerobotic technology.
AVATRINA is a team of researchers from the University of Illinois and Duke University who have designed TRINA, an advanced teleoperated robot designed to work at hospitals [Hauser & Shaw 2020]. This video [Intelligent Motion Lab 2021] shows their telerobotic system in action.
I-Botics is also a participating team in the ANA AVATAR XPRIZE. They are a combination of universities, an applied research organisation, and the high-tech industry from the Netherlands [I-Botics, n.d.]. They recently released a video demo [I-Botics 2021].
The video shows that the operator is in the Netherlands, and the robot is in Norway. They are more than 850 kilometres apart. They use a 3D render of the room in the VR environment, and they overlay an RGB camera feed on top of that environment in synchronous with the operator's head movements.

3 Setup and top-level design

This design uses two computers, an Oculus Quest VR headset and its hand controllers, a UR3 robot arm, an Intel Realsense camera, a USB webcam, a gripper, a servo and an Arduino with a network interface to act as the gripper's controller.
The Intel Realsense camera is on a tripod behind the robot, and the USB webcam is attached on top of the gripper, as shown in Figure 1.
As you can see in Figure 2, there are five major blocks of software represented by the five ellipses.
Program (A) acquires data (position, orientation and button status of hand controllers) from the VR hardware and send them to the remote location over the internet.
Program (B) receives VR data from program (A) and send appropriate commands to the robot and gripper based on the current robot and gripper state. But the commands from this program will first pass through the program (C) for collision checking. Program (B) also sends graphical feedback to program (D) to be sent back to the operator.
Program (C) checks the commands received from program (B) for collision avoidance based on robot state. If there was no collision, then this program will pass these commands to the robot controller. The robot controller will then move the robot accordingly.
Program (D) acquires image data and send them back to the human operator over the internet.
Program (E) receives the image data from program (D) and displays them to the human operator.
Programs (A) and (E) are within the first computer near the human operator. The rest of the programs are within the second computer near the robot.

Figure 1: hardware setup showing the position of the two cameras

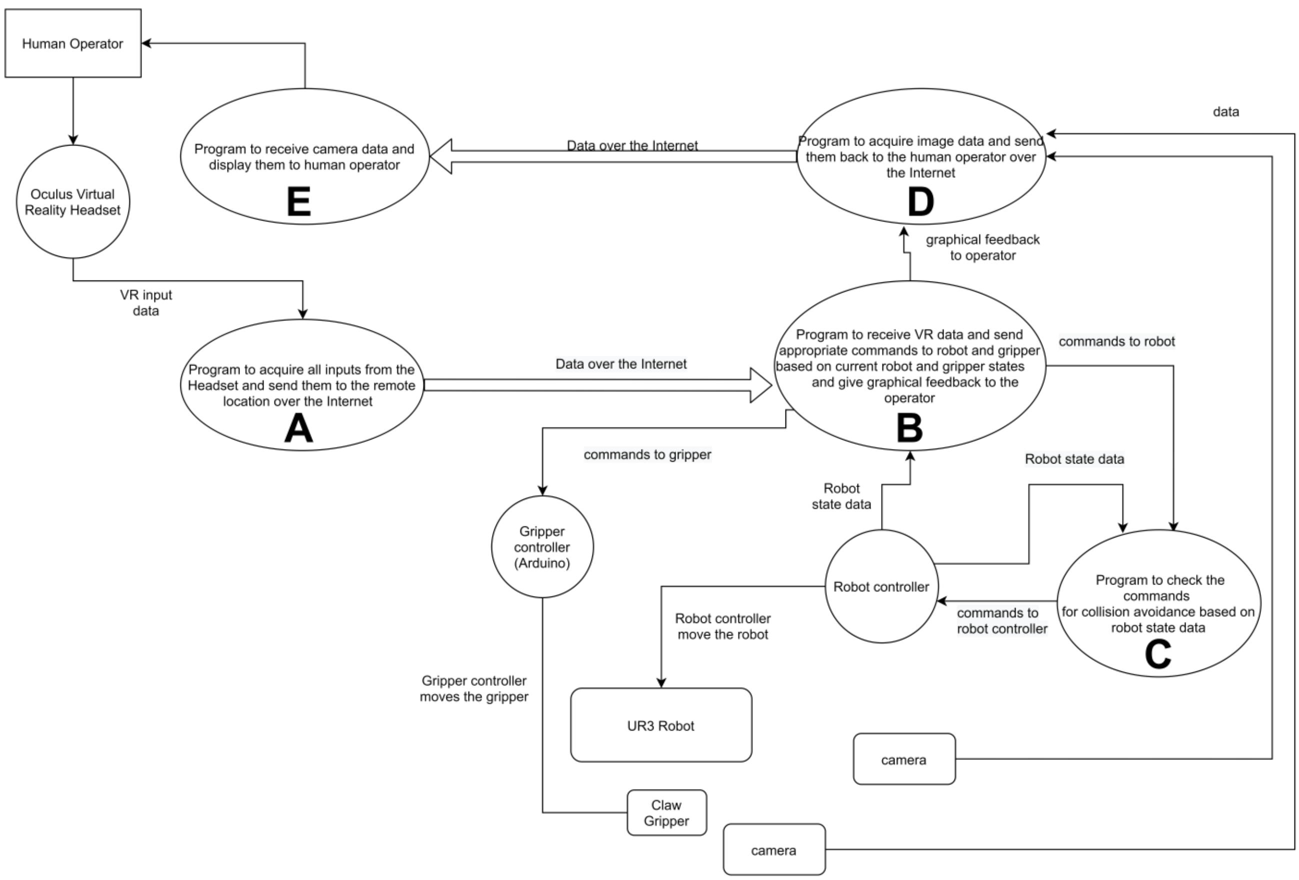

Figure 2: the top-level design of software

4 Control method

4.1 Spatial position mapping

The data that is acquired from the VR hand controller is the status of all buttons, the x, y, z spatial coordinates and the x, y, z, w rotational quaternion of the controller. This version of the VR hardware can provide this data at the rate of 50 Hz. Every data frame will occupy a single UDP packet that is sent to the network address of the computer near the robot.

The UDP protocol is preferred over TCP because it is not important if a handful of control packets are lost in transmission. If the computer near the robot does not receive any command for a while, it will execute the last received command, and then if still no new command is received, it will stop the robot and will wait until a new command is received.

Each control packet has a packet-ID. In the case that the command packets arrive out of order, the computer near the robot will disregard older command packets. It will wait until a packet arrives that has a higher packet-ID number than the last accepted command packet.

The computer near the robot will also receive state data from the robot with a rate of 125 Hz. These include the current angular position and velocity of every joint.

Using the D-H kinematics model, the current spatial coordinates and direction of the robot's end-effector can be calculated. The general D-H kinematics formula is:

$$T_j = T_{j-1}A_j \qquad (1)$$

Here, T_j is the homogeneous transfer matrix of joint j of the robot arm, and T_{j-1} is the homogeneous transfer matrix of joint j-1 of the robot arm. A_j is:

$$A_j = \begin{bmatrix} cos\theta_j & -sin\theta_j cos\alpha_j & sin\theta_j sin\alpha_j & a_j cos\theta_j \\ sin\theta_j & cos\theta_j cos\alpha_j & -cos\theta_j sin\alpha_j & a_j sin\theta_j \\ 0 & sin\alpha_j & cos\alpha_j & d_j \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where θ_j is the joint angle, d_j is the joint link offset, a_j is the joint link length, and α_j is the joint link twist [Corke 2013]. These D-H parameters are specific to each robot arm. While they can be directly measured from the robot arm, they can usually be found in the robot's documentation or other sources.

The UR3 robot arm used in this project has six degrees of freedom. Therefore, to get the coordinates and direction of its end-effecter, T_6 must be calculated.

The origin of the robot's world is set to the base of the robot, which makes T_0 the identity matrix so it can be ignored.

Also, the transfer matrix of the robot tool or gripper in this case is ignored to further simplify the kinematics calculations. Therefore, equation (1) to calculate T_6 simplifies to:

$$T_6 = \prod_{j=1}^{6} A_j = A_1 A_2 A_3 A_4 A_5 A_6 \quad (2)$$

Early on in designing the mapping algorithm, it was realised that because the layout and proportion of joints on the human arm are significantly different to the UR3, mapping directly between the two is very difficult.
The devised solution was to separate linear spatial motion from the rotational motion. This solution was inspired by the layout of joints of the UR3 itself.
The six joints of the UR3 are two shoulder joints, one elbow joint, followed by three wrist joints.
We can infer that the first three joints are responsible for most of the linear spatial motion and the last three joints are mostly responsible for the end-effector's rotational direction. Therefore, it is possible to consider the UR3 as a three-joint robot and map the spatial coordinates of the third joint to the spatial coordinates of the operator's hand controller.
Therefore, instead of T_6, we use T_3, which can be calculated from:

$$T_3 = \prod_{j=1}^{3} A_j = A_1 A_2 A_3 \quad (3)$$

For the actual mapping equation, a PID controller can be used. Although through experimentation, using only the proportional value was proved to be sufficient.
Therefore, the equation for calculating the next robot position based on the movement of the operator's hand becomes:

$$robot\ next\ x = T_{3x} + P\Delta x_{controller} \quad (4)$$

$$robot\ next\ y = T_{3y} + P\Delta y_{controller} \quad (5)$$

$$robot\ next\ z = T_{3z} + P\Delta z_{controller} \quad (6)$$

T_{3x}, T_{3y} and T_{3z} are the x, y, z components of the transform matrix of the third joint T_3. $\Delta x_{controller}$, $\Delta y_{controller}$ and $\Delta z_{controller}$ are the change in the x, y, z coordinates of the operator's hand controller. P is the proportional component of the PID controller.

The robot's next x, y, z values can be used to calculate the inverse kinematics for the first three joints of the UR3 robot.
Here, another benefit of separating spatial and rotational motions becomes apparent as calculating inverse kinematics is easier and faster for a three joint robot compared to a six joint robot.
For calculating the inverse kinematics, one can expand equation (3) and use the T_{3x}, T_{3y} and T_{3z} values calculated in equations (4) to (6) to solve for the three joint angles corresponding to the first three joints of the robot. This will be a system of three non-linear equations and three variables.
Notice that the orientation parameters are not a constraint for this calculation and will be ignored. If there are multiple valid solutions, the solution that is closest to the current robot joint angles will be selected.

4.2 Rotational mapping

Once the first three joint angles are calculated, we can put them back in equation (3) to get a new T_3. Then, convert the quaternion of the operator's hand controller to a homogeneous transfer matrix with the spatial x, y, z elements set to zero. This is only to show that we do not care about its x, y, z coordinates. We call this matrix K, and we write the following equation:

$$K = T_3 A_4 A_5 A_6 \quad (7)$$

Notice that equation (7) is only true for the rotational elements of K, not for its x, y, z spatial coordinates.
We can expand equation (7) for its rotational elements and solve for joint angles four to six. This will be a system of nine non-linear equations with three variables. Again, if there are multiple valid solutions, the solution that is closest to the current robot joint angles will be selected.
One thing to notice is that the mapping for the spatial position is relative because the change in position of the operator's hand controller is used, as seen in equations (4) to (6). However, as you can see in equation (7), the values for the orientation of the operator's hand controller are directly used in the mapping of the robot's end-effector orientation.
This can cause a problem when the remote control is started, and the initial orientation of the operator's hand controller is significantly different from the robot's current end-effector orientation. In that case, the robot's end-effector would suddenly make a large turn to match its orientation with the orientation of the operator's hand controller.
This problem can be mitigated by reducing the speed of wrist rotation and detecting possible collisions using haptic sensors or laser scanners. The user can also be instructed to first sync the orientation of the robot's wrist to the hand controller in an open space before approaching to pick an object.
To the author's knowledge, this method of decoupled mapping and separate control of translation and orientation between a hand controller and a six DOF robot arm is innovative and new in the field of Telerobotics.
The whole inverse kinematics calculation for both position and orientation can run extremely fast at the rate of around 3400Hz. It is bottlenecked by the UR3 robot firmware, which only can accept commands at the maximum rate of 128Hz. This extra time or headroom that the system has can be used to do real-time active collision avoidance.
Currently, collision avoidance is only implemented for self-collision and collision to the ground (table's surface), but it can easily be integrated with a LIDAR or RGBD camera for general collision avoidance.

5 Data flow from the robot to the operator

Currently, the only data that travels back to the operator is the image data. One advantage of this prototype is that the entire video transmission pipeline and protocol is implemented from scratch in C++. This has made it possible to eliminate many intermediate and unnecessary steps and has led to lower transmission delay and latency in video transmission.

Both cameras used have a resolution of 1920 by 1080 pixels. First, each frame is resized down to 1/3 of its size. Some graphics occasionally will be added to these images when needed. For example, when the operator is opening or closing the gripper, the gripper's current position will be shown on the image stream of the camera that is attached to the gripper as seen in Figure 3.
Each frame is compressed to JPEG format. Then the resulting JPEG image is sliced into packets. Each packet can have a maximum size of 1440 bytes. The first 8 bytes of each packet is reserved for the frame identification number. This is a unique integer number that is given to each image frame and can be from 1 to 256^8. The next byte is reserved for the section number.
By doing lots of trials, it was determined that each image will be around 30 to 40 packets and will never be above 255. Therefore, only one byte to represent the section number is needed. The rest of the packet's payload will be the compressed image data.

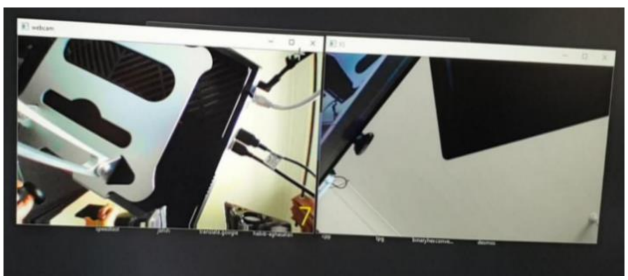

Figure 3: the two camera feeds. Camera feed of the gripper on the left shows the current position of the gripper claws.

6 Experimental results

After completing the system, a remote test was conducted on 31 January 2021. An operator in Western Sydney was connected to a robot on the UTS city campus. As you can see in Figure 4 and in this video, the operator on the left and the robot on the right were more than 44 kilometres apart.
The operator only needed a few minutes of training as the system is highly intuitive and easy to use. Also, a homing position was set to further simplify the picking process.
Two tests were conducted, one with four items and the other with five items. The total time for the two tests was 6 minutes and 53 seconds, which gives an average pick and place time of 45.89 seconds per item [Aghasafari, 2021]. This video is in real-time and has not been sped up in any way and clearly demonstrates that this telerobotic system can be used to remotely pick and place small objects from a 44-kilometre distance.

7 Contributions and Conclusion

In this paper, an innovative, simple, yet highly effective way of mapping between the UR3 robot arm and a VR hand controller was introduced. This mapping was used to develop an intuitive telerobotic control system that is highly effective in performing remote pick and place of small objects.
The advantage of this mapping method is three-fold:
First, inverse kinematics becomes much simpler and faster to calculate because instead of calculating inverse kinematics for a six-DOF arm, you can calculate it for two three-DOF arms that are attached to each other.
Second, this method makes the movement of the arm very predictable and intuitive as there is no trajectory planning involved. Both spatial position and orientation will react one to one to the movement of the user's hand controller.

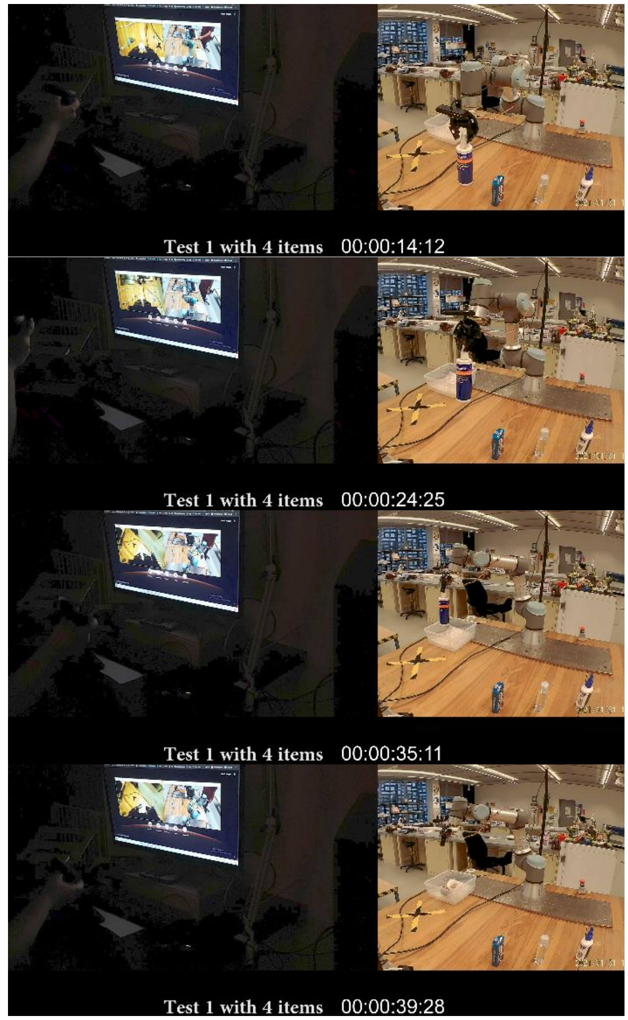

Figure 4: The telerobotic experiment. The operator on the left is 44 kilometres away from the robot on the right.

Both the [Lipton et al. 2017] and [Gharaybeh et al. 2019] systems use a motion planner that performs the inverse kinematics and trajectory planning over the whole six joints of the arm. This may introduce motions that deviate from what the user expects based on the motion of their hand. This is because the layout of joints of the human hand is different from the layout of the robot's joints.
This problem can be avoided if the target position and orientation are not coupled together. Half of the robot's joints can follow the position, and the other half can follow the orientation of the user's hand and produce a combined motion that is one to one relative to the user's hand motion which will be an extremely intuitive experience for the user.

Third, this system can easily be replicated for other robot arms. For a robot arm with six or seven joints, the last three joints can be reserved for orientation movements, and the rest of the joints can be dedicated to spatial position movement.
Also, the user does not need to interact with virtual objects like [Lipton et al. 2017] to control the robot arm movements. The user can naturally move their hand in the space, and the robot follows their hand motion.
The code for this prototype is implemented mainly in C++ and MATLAB. This prototype has made significant improvements to data processing and transmission delay. The camera feeds run at 30Hz, and the VR tracking data feeds run at 50Hz over the internet and 44-kilometre distance. In comparison, the data rate of [Lipton et al. 2017] over the internet was 5Hz and for [Gharaybeh et al. 2019] was 0.016Hz.
The experimental results have proven that the innovative control method introduced in this paper is highly effective. They also prove that consumer VR hardware such as Oculus Quest can be used to perform remote pick and place of small objects.
In conclusion, Telerobotics is a viable path for bringing robots to everyday life. It has a huge potential and can make the world a better place.

8 Further developments

There are many areas that are planned for further development in future. Currently, the VR headset needs to be connected to a computer. The program on this computer can read the tracking data from the headset. The camera streams are displayed on two windows on this computer, and the user can display these two windows inside of the VR environment using the Oculus Virtual Desktop software.
By designing a standalone VR game, the user will no longer need a separate computer and can only use the VR headset. Also, this will make it possible to design a more intuitive and more immersive experience for the user.
Currently, there is no transferring of sound from the robot side to the operator side. This is also an area that can help to make a more immersive experience for the user.
Additionally, the collision avoidance software will further be developed to include dynamic objects and obstacles.
These software improvements should increase the performance and safety of the whole telerobotic system.
In addition to the software improvements, it is desirable to experiment with better hardware such as multiple robot arms, mobile robot base, bionic hand grippers, haptic gloves and more.

References

[Aghasafari, 2021] *Habib Robotics first remote pick and place test*, YouTube, <https://youtu.be/q229ROo3SK0>.

[Corke, 2013] *Robotics Vision and Control Fundamental Algorithms in MATLAB*, second edition, Springer-Verlag, Berlin Heidelberg, pp. 138-139.

[De Pace et al. 2021] F. De Pace, G. Gorjup, H. Bai, A. Sanna, M. Liarokapis, M. Billinghurst. Leveraging Enhanced Virtual Reality Methods and Environments for Efficient, Intuitive, and Immersive Teleoperation of Robots. *ICRA 2021*.

[Franzluebbers & Johnsen 2019] A. Franzluebbers, K. Johnsen. Remote Robotic Arm Teleoperation through Virtual Reality. *SUI '19: Symposium on Spatial User Interaction,* Article No. 27, pp. 1-2.

[Gharaybeh et al. 2019] Z. Gharaybeh, H. Chizeck and A. Stewart, Telerobotic Control in Virtual Reality, *OCEANS 2019 MTS/IEEE SEATTLE*, pp. 1-8.

[Hauser K. & Shaw R. 2020] *How Medical Robots Will Help Treat Patients in Future Outbreaks*, IEEE Spectrum, <https://spectrum.ieee.org/automaton/robotics/medical-robots/medical-robots-future-outbreakresponse>.

[I-Botics, n.d.] *ANA AVATAR XPRIZE*, < https://i-botics.com/projects/xprize/ >.

[I-Botics 2021] *i-Botics XPRIZE - Semifinalist Selection Submission – Full*, YouTube, <https://youtu.be/ddcS92uzGCc>.

[Intelligent Motion Lab 2021] *ANA Avatar XPRIZE semifinals selection video: Dr. TRINA*, YouTube <https://youtu.be/G2yamXSizDQ>.

[Klamt et al. 2019] T. Klamt, M. Schwarz, C. Lenz, L. Baccelliere, D. Buongiorno, T. Cichon, A. Di Guardo, D. Droeschel, M. Gabardi, M. Kamedula, N. Kashiri, A. Laurenzi, D. Leonardis, L. Muratore, D. Pavlichenko, A. S. Periyasamy, D. Rodriguez, M. Solazzi, A. Frisoli, M. Gustmann, J.n Roßmann, U. Süss, N. G. Tsagarakis, S. Behnke. Remote Mobile Manipulation with the Centauro Robot: Full-body Telepresence and Autonomous Operator Assistance. *Journal of Field Robotics (JFR), Wiley,* 37(5), 889-919.

[Lipton et al. 2017] J. I. Lipton, A. J. Fay and D. Rus, "Baxter's Homunculus: Virtual Reality Spaces for Teleoperation in Manufacturing," in IEEE Robotics and Automation Letters, vol. 3, no. 1, pp. 179-186, Jan. 2018, doi: 10.1109/LRA.2017.2737046

[MITCSAIL 2017] *Operating Robots with Virtual Reality*, YouTube, <https://youtu.be/4a-W3Od5-t8>.

[Oculus n.d.] Oculus Quest 2, < https://www.oculus.com/quest-2/ >

[Van de Merwe et al. 2019] D. B. Van de Merwe, L. Van Maanen, F. B. Ter Haar, R. J. E. Van Dijk, N. Hoeba, N. Van der Stap. Human-Robot Interaction During Virtual Reality Mediated Teleoperation: How Environment Information Affects Spatial Task Performance and Operator Situation Awareness. *International Conference on Human-Computer Interaction*

[Verd´ın et al. 2021] R. Verd´ın, G. Ram´ırez, C. Rivera, G. Flores. Teleoperated aerial manipulator and its avatar. Communication, system's interconnection, and virtual world. *ICUAS 2021* Athens, Greece.

[Xprize 2021] *ANYWHERE IS POSSIBLE,* Xprize.org, <https://www.xprize.org/prizes/avatar>.