Max B. Schäfer*, Kent W. Stewart, Nico Lösch and Peter P. Pott

# Telemanipulation of an Articulated Robotic Arm using a Commercial Virtual Reality Controller

**Abstract:** Access to systems for robot-assisted surgery is limited due to high costs. To enable widespread use, numerous issues have to be addressed to improve and/or simplify their components. Current systems commonly use universal linkage-based input devices, and only a few application-oriented and specialized designs are used. A versatile virtual reality controller is proposed as an alternative input device for the control of a seven degree of freedom articulated robotic arm. The real-time capabilities of the setup, replicating a system for robot-assisted teleoperated surgery, are investigated to assess suitability. Image-based assessment showed a considerable system latency of $81.7 \pm 27.7$ ms. However, due to its versatility, the virtual reality controller is a promising alternative to current input devices for research around medical telemanipulation systems.

**Keywords:** Teleoperated Robot-Assisted Surgery, RAS, Industrial Robot, Virtual Reality, Input Device, Low-Cost

https://doi.org/10.1515/cdbme-2020-3033

## 1 Introduction

Recently, a strong emergence in robot-assisted surgery (RAS) can be observed [1]. As a subcategory of RAS, teleoperated minimally invasive surgery (MIS) provides numerous advantages compared to conventional MIS [2, 3]. This includes, improved ergonomics, intuitive handling, tremor filtration, and the possibility to scale motions. However, the widespread use of available systems is limited, which is likely caused by high acquisition and operating costs [4, 5]. To enable frequent use of such systems, numerous issues have to be addressed to improve and/or simplify their components.

Commercial virtual reality (VR) controllers have already been used in research projects for precise control, such as teleoperation during repair in space [6] or for an accessible platform for education and research around VR teleoperation in RAS [7]. VR controllers in these research projects were predominantly selected due to their high versatility with six degrees of freedom, minimal hardware requirements, and the possibility of free motion in a relatively big working space. Furthermore, due to the demand of VR systems in the consumer gaming market and the resulting competitive fair price, the application of commercial systems such as the Oculus Rift (Oculus VR, Facebook Technologies LLC., CA, USA) and the HTC Vive (High Tech Computer Corporation, Taoyuan, TW) for scientific studies has become more feasible [8].

In this paper, an unconventional and versatile approach for the input device of a telemanipulation system is investigated. An HTC Vive VR controller is used as a freely movable and handheld device to interact with an industrial robotic manipulator suited for RAS [9]. The relatively unrestricted motion possibilities of the VR controller enable future investigations around user interactions in RAS such as the effect of motion scaling on intuitiveness and precision. Further, the real-time capabilities of a commercial VR system in combination with an industrial articulated robotic manipulator (Panda, Franka Emika GmbH, Munich, GER), replicating a RAS system for teleoperated surgery, are assessed.

## 2 Methods and Setup

### 2.1 Robot Manipulator

A seven degree of freedom articulated robotic arm with a parallel gripper as endeffector was used as the manipulator for the setup of a master-slave telemanipulation system. The robotic arm features a maximum payload of 3 kg, a sufficient range of motion, and a 1 kHz control loop. Due to its seven

***Corresponding author: Max B. Schäfer:** Institute of Medical Device Technology, University of Stuttgart, Pfaffenwaldring 9, Stuttgart, Germany, max.schaefer@imt.uni-stuttgart.de
**Kent W. Stewart, Nico Lösch, Peter P. Pott:** Institute of Medical Device Technology, University of Stuttgart, Germany

joints, the robotic arm is kinematically redundant, which allows motion in the elbow joint whilst the endeffector stays at a fixed pose. The robot's control unit is directly connected to a workstation by TCP/IP. The latter is equipped with an Intel®Core™ i7-8700K CPU (Intel Corporation, Santa Clara, CA, USA), 16 GB of DDR4 2667 MHz RAM and a Nvidia GeForce RTX 2080 graphics processing unit (Nvidia Corporation, Santa Clara, CA, USA).

Two external safety buttons were present. The user stop button interrupts the execution of the running control loop software-wise, and an emergency stop button creates an electronic and mechanical shutdown.

A C++ Application Programming Interface (API) named libfranka is provided by the robot's manufacturer. The library directly communicates with the Franka Control Interface running on the robot's control unit and provides interfaces for processing real-time and non-real-time commands. This library can also be used to obtain robot model parameters such as the Jacobian matrix or the mass of the robot, and accesses information about the robot's state to get torque sensor data in real-time.

## 2.2 Virtual Reality Controller

An HTC Vive VR controller was used as the input device for the telemanipulation setup. Two stationary sensor units continuously track the controller at a 60 Hz refresh rate, per unit. By sending out vertical and horizontal pulses of infrared light in an alternating manner, an angle of 120° per cycle is covered in each direction. To update the current tracked position, path integration is used.

To access the SteamVR runtime (Valve Corporation, Bellevue, WA, USA), the OpenVR Software Development Kit (Valve Corporation, Bellevue, WA, USA) was utilized. A C++ library, referred to as *VRControl*, was developed based on the examples provided by the OpenVR Software Development Kit. The setup of both, robot manipulator and VR input device, are shown in Fig. 1.

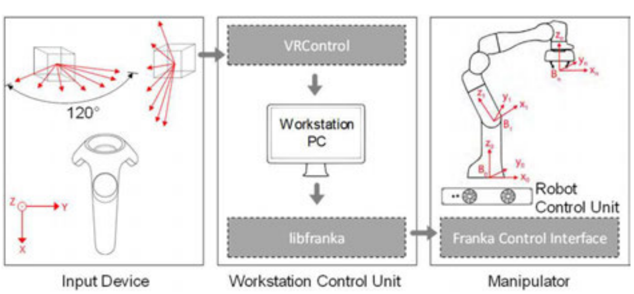


**Figure 1:** Tracking of the HTC VR input device performed by two sensor units. The tracked device pose is then transferred via a workstation PC to the Franka Emika.

## 2.3 Control

The present control algorithm is implemented as a task space force controller which precisely controls the endeffector. Force control is required since forces need to be exerted and compensated for such as during interaction with tissue in RAS [10, 11]. The force control is implemented as an impedance controller consisting of a virtual mass-spring-damper system represented by a set of linear and non-linear second-order differential equations controlling the endeffector position in Cartesian space. So, the force is controlled by controlling position since no force error feedback loop is present.

The control input $\boldsymbol{u} \in \mathbb{R}^{m\times 1}$ ($m$ is the number of degrees of freedom) is determined in two steps to achieve the desired dynamic characteristic for the interaction between manipulator and environment. In the first step, decoupling and linearization of the closed-loop dynamics is performed in the task space. In the second step, an impedance model described by a second-order mechanical system dynamically balances the contact forces present at the endeffector. Without the possibility of force feedback, Eq. 2 describes the final control strategy which resembles a task space proportional-derivative controller with gravity compensation [10–12]:

$$\boldsymbol{u}(t) = \boldsymbol{J}_a^T(\boldsymbol{q})\left(\boldsymbol{K}_m\left(\boldsymbol{x}_d(t) - \boldsymbol{x}(t)\right) + \boldsymbol{D}_m\left(\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}(t)\right)\right) + \boldsymbol{g}(\boldsymbol{q}) \quad (2)$$

where $\boldsymbol{q}$ is the vector of joint variables/angles, $\boldsymbol{J}_a^T(q) \in \mathbb{R}^{m\times 1}$ is the analytical Jacobian, $\boldsymbol{K}_m$ and $\boldsymbol{D}_m$ act as virtual stiffness and damping matrices, $\boldsymbol{x}$ is the actual endeffector position, $\boldsymbol{x}_d$ the desired endeffector position and $\boldsymbol{g} \in \mathbb{R}^{m\times 1}$ is the gravitational joint torque.

Regarding the implementation with respect to the Panda robot, the approach of Eq. 2 was used. Gravity and Coriolis compensation can be obtained from the respective control library. Therefore, Eq. 2 is reduced to

$$\boldsymbol{u}(t) = \boldsymbol{J}_a^T(q)\left(\boldsymbol{K}_m\left(\boldsymbol{x}_d(t) - \boldsymbol{x}(t)\right) + \boldsymbol{D}_m\left(\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}(t)\right)\right) \quad (3)$$

with $\boldsymbol{x}_d \neq 0$ due to non-quasi-static behavior, because the robot is in motion. The actual position and orientation, $\boldsymbol{x}$, are provided by the robot. The Jacobian $\boldsymbol{J}_a$ can be accessed through the control library depending on the current state of the robot. $\boldsymbol{K}_m$ and $\boldsymbol{D}_m$ are determined by assuming a harmonic oscillator with damping ratio of 1 (critically damped) and a mass of 1 kg (approximate endeffector mass), which results in the following Equation:

$$\boldsymbol{D}_m = 2\sqrt{\boldsymbol{K}_m} \quad (4)$$

A stiffness of 600 N·m$^{-1}$ was selected in order to guarantee a short response time while still showing a stable behavior in the respective workspace.

### 2.4 System Evaluation

The overall system latency and robot controller latency was assessed. Overall system latency was investigated using visual markers on the VR controller and the robotic endeffector. Both the robot endeffector and VR controller were placed in front of a white screen within the field of view of the measurement camera. Using the slow-motion camera of the Apple iPhone X (Apple Corporation, Cupertino, CA, USA), the delay between initial motion of both markers was determined by the camera image frames. Recording with a sampling rate of 240 Hz, a time resolution of 4.2 ms was achieved. Note, the control loop of the robot was modified to only allow movement of the robot in one direction, and the same latency was assumed for all possible translations and rotations. An overall number of five measurements was carried out and the average delay (latency) determined.

The robot control algorithm delay was assessed by applying two different synthetic input functions, in software, to the robot controller. A step function and a raised cosine function (1 s period time) were chosen, both having an amplitude of 100 mm. The cosine function was used to resemble a more natural human movement. The response time was determined by the onset latency of the robot's position feedback, given by the time it took the robot to respond to an input signal and achieve a threshold movement of 0.2 mm (due to the positioning accuracy of the robot [13]).

## 3 System Design

One handheld controller of the VR system was used to control the single endeffector of the robot manipulator. The *VRcontrol* routine obtains a homogeneous matrix, containing translation as well as rotation of the input handle. This pose is transformed to the base frame of the robot and forwarded as its new desired pose. The robot control algorithm generates the respective control arguments from the desired pose. The specified control arguments are processed by a low-pass filter with a cutoff frequency of 100 Hz to diminish high frequency components of the input signal and a rate-limiter, which ensures the adherence of inbuilt limits, such as acceleration and velocity constraints. For translation, a velocity of up to 1.7 m/s and an acceleration of up to 13 m/s$^2$ was possible, whereas for rotation, a velocity of 2.5 rad/s and an acceleration of 25.0 rad/s$^2$ was set as the upper limit. The filtered and verified signals of the motion generator are then processed by the robot kinematics completion, transforming them into torques, which can be applied to the joints of the robot. Once each joint is actuated, the robot returns the robot state, which contains information about the robot, including joint positions and velocities, the position, orientation, and twist of the endeffector frame and the actual applied torques of each joint.

## 4 Results

### 4.1 System Evaluation

Considering the overall system response time, a mean latency of 81.7 ± 27.7 ms was determined based on the camera image frames. The control algorithm of the robot thereby showed a mean onset latency of 12.0 ± 0.0 ms for the step input and 70.2 ± 1.3 ms for the raised cosine input function (see Fig. 2).

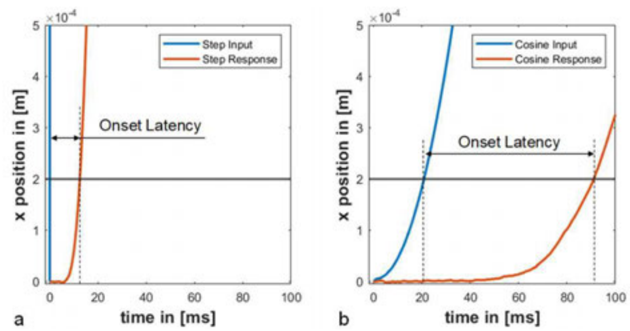


**Figure 2:** Exemplary response and onset latency to a step input function (a) and to a raised cosine input function (b) using a threshold of 0.2 mm.

## 5 Discussion

The results show an initial setup towards a telemanipulation system for research around RAS. Focus was on the usage of a commercial virtual reality controller as a versatile input device to control a robotic manipulator. The functionalities of the setup mainly depend on the chosen control strategy. The need to physically interact with the environment reduces the suitable control strategies. De Wit et al. suggest a parallel control algorithm with a force feedback loop and the possibility of force regulation or the usage of a hybrid force/motion controller with the introduction of constraints [11]. However, previous work shows that simple controllers such as the PD and proportional-integral-derivative (PID) feedback are effective for setpoint control despite the nonlinearity und uncertainty of the robot dynamics [14]. This supports the control attempt used in this study but does not impair the necessity of further investigation into the control strategy. In addition, future control strategy could take advantage of the torque sensors present in each of the robot's joints to enable better force feedback.

The measured overall system response time (81.7 ± 27.7 ms) is considerable, but still in a reasonable range for surgical tasks, since the effect of latency is manageable up to 300 ms for experienced surgeons [15]. The high standard deviation of the overall system latency is caused by the limited spatial and temporal resolution of the camera used. Further, the assumption of having the same latency for translation and rotation need to be reconsidered since different velocity and acceleration limits occur. Considering only the robot control algorithm, the PD controller seems to be adequately tuned for the first proof of concept. However, latencies could differ with varying input since PD control is dependent on the size of the input signal. The variation in the response time to a step input showed a standard deviation of zero, compared to the cosine input which had a standard deviation of ± 1.3 ms. This could be caused by static friction in the robot's joints having less of an effect when the input changes rapidly in comparison to a slow input such as a cosine, leading to more consistent behaviour. However, the control algorithm of the robot must be further investigated and improved, since the occurring response times significantly contribute to the overall system latency, potentially compromising intuitive use [16]. Since the measured latencies for the overall system show considerable standard deviations, no exact prediction of the VR system performance can be made. Niehorster et al. quantify the end-to-end latency of the HTC Vive tracking system to be around 22 ms [8], which also contributes considerably to the total system latency. However, the developed library of the VR system can be optimized regarding faster and more efficient calculations and matrix manipulations.

# 6 Conclusion

A telemanipulation system for research around robot-assisted surgery was set up using low-cost and commercially available components. The chosen easily accessible commercial VR controller is a promising alternative to current input devices for medical telemanipulation systems. Latency, more sophisticated control strategies, and occlusion robust tracking sensors should be assessed in the future.

**Author Statement**

Research funding: The author state no funding involved. Conflict of interest: Authors state no conflict of interest. Informed consent: Informed consent is not applicable. Ethical approval: The conducted research is not related to either human or animals use.

## References

[1] D. Ranev and J. Teixeira, "History of Computer-Assisted Surgery," *The Surgical clinics of North America*, vol. 100, no. 2, pp. 209–218, 2020.

[2] R. H. van der Schatte Olivier, C. D. P. Van't Hullenaar, J. P. Ruurda, and I. A. M. J. Broeders, "Ergonomics, user comfort, and performance in standard and robot-assisted laparoscopic surgery," *Surgical endoscopy*, vol. 23, no. 6, pp. 1365–1371, 2009.

[3] S. M. Prasad, S. M. Prasad, H. S. Maniar, C. Chu, R. B. Schuessler, and R. J. Damiano, "Surgical robotics: impact of motion scaling on task performance," *Journal of the American College of Surgeons*, vol. 199, no. 6, pp. 863–868, 2004.

[4] C. Bolenz *et al.,* "Cost comparison of robotic, laparoscopic, and open radical prostatectomy for prostate cancer," *European urology*, vol. 57, no. 3, pp. 453–458, 2010.

[5] R. M. Higgins, M. J. Frelich, M. E. Bosler, and J. C. Gould, "Cost analysis of robotic versus laparoscopic general surgery procedures," *Surgical endoscopy*, vol. 31, no. 1, pp. 185–192, 2017.

[6] H. Wang, R. Mecham, and B. Zhang, "A Method Targeting Repair in Space: Tele-operating a Collaborative Robot with Virtual Reality," in *IEEE-CYBER: The 7th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, Honolulu, HI, 2017, pp. 1068–1071.

[7] N. Sanchez-Tamayo and J. P. Wachs, "Collaborative Robots in Surgical Research," in *HRI'18 companion, Chicago, IL, USA*, Chicago, IL, USA, 2018, pp. 231–232.

[8] D. C. Niehorster, L. Li, and M. Lappe, "The Accuracy and Precision of Position and Orientation Tracking in the HTC Vive Virtual Reality System for Scientific Research," *i-Perception*, vol. 8, no. 3, 2017.

[9] M. B. Schäfer, K. W. Stewart, and P. P. Pott, "Industrial robots for teleoperated surgery – a systematic review of existing approaches," *Current Directions in Biomedical Engineering*, vol. 5, no. 1, pp. 153–156, 2019.

[10] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, planning and control*. London: Springer, 2010.

[11] C. C. Wit, B. Siciliano, and G. Bastin, *Theory of Robot Control*. London: Springer, 1996.

[12] P. Song, Y. Yu, and X. Zhang, "Impedance Control of Robots: An Overview," in *2017 2nd International Conference on Cybernetics, Robotics and Control: CRC: Chengdu, Sichuan, China*, Chengdu, 2017, pp. 51–55.

[13] Franka Emika GmbH, "Panda - Datasheet," Munich, 2019.

[14] S. Arimoto, *Control theory of non-linear mechanical systems: A passivity-based and circuit-theoretic approach*. Oxford: Clarendon Press, 2003.

[15] M. Anvari *et al.,* "The impact of latency on surgical precision and task completion during robotic-assisted remote telepresence surgery," *Computer aided surgery : official journal of the International Society for Computer Aided Surgery*, vol. 10, no. 2, pp. 93–99, 2005.

[16] M. B. Schäfer, K. W. Stewart, N. Lösch, and P. P. Pott, "Assessment of a Commercial Virtual Reality Controller for Telemanipulation of an Articulated Robotic Arm," *in 2020 8th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob), New York*, 2020. Pending Acceptance.