Cobity: A Plug-And-Play Toolbox to Deliver Haptics in Virtual Reality

Steeven Villa
LMU Munich
Munich, Germany
steeven.villa@ifi.lmu.de

Sven Mayer
LMU Munich
Munich, Germany
info@sven-mayer.com

ABSTRACT

Haptics increase the presence in virtual reality applications. However, providing room-scale haptics is an open challenge. Cobots (robotic systems that are safe for human use) are a promising approach, requiring in-depth engineering skills. Control is done on a low abstraction level and requires complex procedures and implementations. In contrast, 3D tools such as Unity allow to quickly prototype a wide range of environments for which cobots could deliver haptic feedback. To overcome this disconnect, we present Cobity, an open-source plug-and-play solution to control the cobot using the virtual environment, enabling fast prototyping of a wide range of haptic experiences. We present a Unity plugin that allows controlling the cobot using the end-effector's target pose (cartesian position and angles); the values are then converted into velocities and streamed to the cobot inverse kinematic solver using a specially designed C++ library. Our results show that Cobity enables rapid prototyping with high precision for haptics. We argue that Cobity simplifies the creation of a wide range of haptic feedback applications enabling designers and researchers in human-computer interaction without robotics experience to quickly prototype virtual reality experiences with haptic sensations. We highlight this potential by presenting four different showcases.

CCS CONCEPTS

• **Human-centered computing** → *Virtual reality*; • **Computer systems organization** → *Robotics.*

KEYWORDS

robots, cobots, virtual reality, toolbox, unity, haptics

ACM Reference Format:
Steeven Villa and Sven Mayer. 2022. Cobity: A Plug-And-Play Toolbox to Deliver Haptics in Virtual Reality. In *Mensch und Computer 2022 (MuC '22), September 4–7, 2022, Darmstadt, Germany*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3543758.3543775

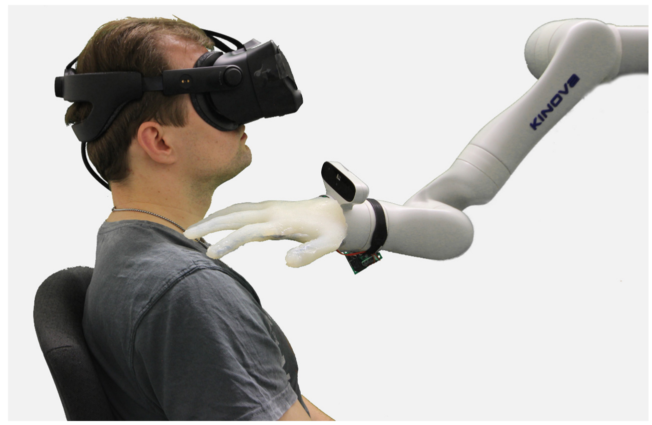


Figure 1: The Social Touch example; we used Cobity to deliver human like touch sensations in virtual reality.

1 INTRODUCTION

Cobots are becoming less expensive and their capabilities more sophisticated, e.g., various attachments. The recent rise in cobot offer makes them a feasible option for researchers and practitioners to use them beyond traditional domains such as manufacturing or assembly [23]. Traditionally, robots that shared a working environment with humans required caging systems and strict operational protocols to safeguard human integrity. However, cobots are safe to interact with in the same space as humans and can perform collaborative tasks. Emerging research use cobots to support humans' day-to-day tasks, e.g., cooking [6, 31]. This uncovers a large range of applications that have previously rarely been explored, such as using cobots in combination with virtual reality (VR).

Researchers and developers have neglected haptics in VR given its technical intricacies; therefore, they did not exploit the full potential of VR. Haptics is inherently a contact experience and requires close interaction with the device. The safety, versatility, and significant reach of commercial cobots pose them as an interesting alternative to the traditional specialized force-feedback haptic devices. In fact, haptic researchers have been steadily reporting new techniques to exploit the potential of cobots in the haptic domain; for example, encountered-type haptics [21], texture rendering [25] or on-demand tangibles. However, controlling cobots is difficult, cf. [16, 28] and thus, most implementations require extensive expertise and time. A common approach is pre-recording motions; however, this is not an option for collaborative tasks as the cobot needs to react to the users' actions.

Researchers and developers proposed numerous solutions to control robots and deliver haptic feedback in VR, such as encounter-type haptics [25], mid-air extended haptics [8], tangible object manipulation [24], and social touch [15]. However, while such sensations can be created it is a time-consuming development process,

MuC '22, September 4–7, 2022, Darmstadt, Germany

ACM ISBN 978-1-4503-9690-5/22/09…$15.00
https://doi.org/10.1145/3543758.3543775

which is not feasible for rapid prototyping. Thus, creating VR applications with cobots requires a lot of time-consuming overhead. Moreover, the technical knowledge that is required hinders people from using cobots to deliver haptic feedback.

To lower the technical burden of using cobots in VR environments, we present Cobity (Cobot + Unity), a new open-source solution to use a virtual simulation environment to control the cobot directly. Here, we use the VR environment as input for the cobots end-effector. This allows practitioners and developers to use one tool, in our case Unity, to control the VR environment the users visually perceive but also the robot which delivers the haptic feedback to the user. In detail, first, we develop a real-time C++ library functioning as a bridge between the cobot and the VR environment. This enables us to build a Unity plugin that allows mapping a 3D object to the end-effector of the cobot. In this way, we transfer the in-game motion of the 3D object onto the real cobot motions. Here, we use an Inverse Kinematics (IK) solver to update the position of the cobot iteratively.

With Cobity, we present a dynamic library allowing the use of Unity for rapid cobot control, enabling prototyping. We run a technical evaluation of our plug-and-play solution, showing precise repeatability and stability. Finally, Cobity supports developers in creating a comprehensive set of different haptic feedback sensations; thus, allowing for a wide range of applications.

2 RELATED WORK

Advances in robotics, especially in robots that are safe for interacting with humans, make it feasible to create novel forms of haptic feedback. In this section, we focus on the current approaches to controlling a cobot. In particular, we assess the utility and usability of the available tools and middleware. We look in particular at how this can be achieved from Unity. We also provide a brief overview outlining the relevance of robotics for haptics and VR.

2.1 Robotic Middle-Ware

A common approach in robot control development is to use robotic middle-ware, especially ROS (Robotic Operatic System) [34]. Such a platform has boosted the growth of robotic applications [20] and allowed the robotics community to share and reuse code in different scenarios [10, 20, 22, 37]. However, such re-usability and robustness come with a cost: Implementing simple applications requires a considerable number of concepts and configurations and, thus, needs time and cognitive resources to get started [18].

The greater availability of cobot allows interaction/game designers to now use them. The need for engineering expertise limits the applicability of the approaches, especially for designers and HCI researchers and practitioners. Several approaches have been proposed to facilitate the communication between graphic engines and robots [1, 3, 40]; of particular relevance, Unity maintains a repository that allows Robotic and Control researchers to simulate their setups in the 3d Engine [40]. Yet, setting up and running this library still requires all the standard processes of setting up a ROS architecture. Therefore, it is better suited to users knowledgeable in the ROS standard to include Unity in their workflow than the other way around. Similarly, Crick et al. [9] proposed ROSBridge to facilitate the development of robotic applications to non-experts by adding an abstraction layer to handle the communication with the ROS middle-ware, particularly for web interfaces. However, it still does not represent a plug-and-play solution for HCI practitioners since it is an add on to the middle-ware but not a solution for communicating directly with the robots.

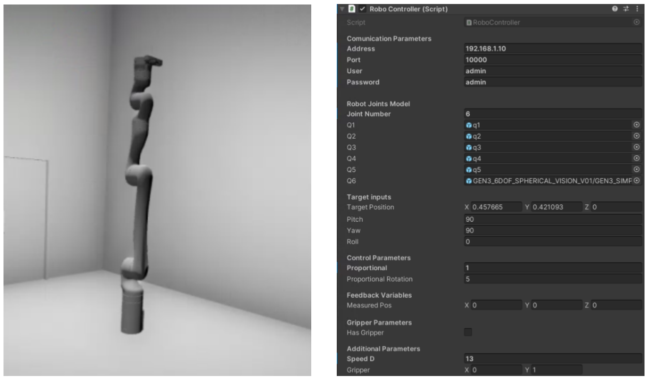
Notably, Bartneck et al. [4] addresses this by proposing an open-source platform that runs as a plugin in Unity and allows prototyping robotic applications; similarly, Günther et al. [14] reported a rapid prototyping platform with support for unity for integrating hardware elements. However, these plugins are focused on Arduino-based projects and do not extend to the usage of manipulators. Thus, Cobity emerges as an alternative that enables HCI and haptic designers to quickly prototype experiences without installing additional platforms and developing complex communication structures. Furthermore, it runs directly in Unity and does not require extensive robotics experience to get started.

2.2 Haptics in VR

VR evolved in recent years due to developments in screen technologies, 3D rendering spatial audio spatial tracking, among others [36]. After decades of development, VR devices moved from research contexts to consumer usage, such as the Oculus Quest[1] by Facebook or HTC Vive Cosmos[2]. Yet, the consumer market for VR orbits around audio-visual stimulation and spatial interaction; further cues from the virtual world are missing, and the haptic community still needs to articulate an ultimate haptic device able to render a wide range of tactile sensations [42]. Although this ultimate device seems far from existing, strategies such as introducing tangible props [29] or vibrotactile stimulation can partially supply the missing sensory content [30]. In parallel, we face a rapid development in robotics, namely the use of cobots is rapidly introduced to middle and small industries [5, 27] and further investigated in the home context [41]. Such technology is closer to the end-user than ever before and is rapidly enabling a wide range of applications relevant to HCI [19]. In the past, being of exclusive use of assembly lines, nowadays cobots are also relevant as assistive devices, kitchen support, cinematic production. We argue that haptics can take advantage of such a revolution and focus its efforts on exploring the design space and possibilities that are now possible thanks to this evolution. Recently presented applications have shown the flexibility of cobot-based haptic interfaces, such as rendering textures, shapes, and kinesthetic feedback [24, 25]. For other application cases of robot manipulators, the reader can refer to Mercado et al. [25, 26]. Finally, Bouzbib et al. [7] who recently compiled an excellent overview on haptic devices for VR. They summarized that haptics can strongly benefit from manipulator-based applications as a step forward towards a flexible and high-quality haptic interface that can be adopted by a broader range of users; envisioning that future, we present a tool that enables non-robotic experts to explore and implement their own haptic applications by bringing a plug-and-play software piece that allows them to control the robot from a more familiar environment (Unity) instead of moving them to unfamiliar paradigms, e.g., ROS interfaces.

[1] https://www.oculus.com/quest-2/

[2] https://www.vive.com/eu/product/vive-cosmos/features/



(a) Robot 3D Model

(b) Plugin Interface

Figure 2: The two visual components of our plugin.

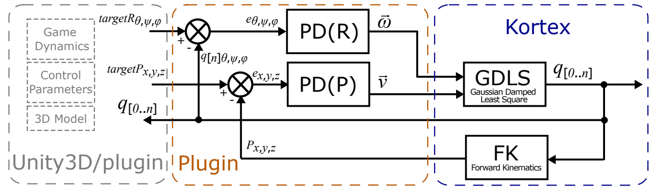


Figure 3: Plugin communication scheme, a C++ dynamic library manages the information exchange between the graphic engine and the control loop of the robot: The graphic engine manages the positions in the virtual environment, the target positions and measured feedback are sent and retrieved from the plugin. The plugin then communicates with the robot loops using the Kortex API to manipulate the robot's end effector towards the target position received from the virtual environment. Also, feedback from the joints and estimated end-effector position are sent back to the virtual environment.

3 TOWARDS PLUG-AND-PLAY ROBOTICS

We developed the Cobity plugin as a bridge to bring robotics closer to HCI researchers, therefore prioritizing easiness to use and setup. The plugin runs in the same operating system (Windows) and does not require network communication with auxiliary platforms. It does not need to launch additional software to establish the connection and communicate with the robot. In the following, we present the details of our implementation.

3.1 Architecture

To reduce computational costs in the host computer and the complexity of the communication setup, we aimed to provide direct communication from the graphic engine to the robot joints. However, such implementation raises a series of conflicts, such as the control frequency required to make the robot movement smooth and stable. A direct control loop over the robot joints involves a minimum of 1Khz to avoid instabilities and oscillations in the motor; however, a graphic engine such as unity typically runs at ~30 to 120 fps. While this is frequently acceptable for graphics, it's not suitable for robot control. Therefore, we developed a dynamic library that manages this communication by running a velocity control over the 1KHz loop of the robot, see Figure 3. This introduces a middle loop that sets targets to the High-frequency loop whenever it reads them from the physics loop in the graphic engine. The central loop reads positions in coordinates centered in the robot's base. The cartesian coordinates are converted into velocity vectors (Rotational and Translational), communicated to the kinematics loop, and finally applied to the joint motors. The feedback from the encoders is read from the encoders in the robot and then sent to the graphic engine to animate the cobot's model.

3.2 Control

We implemented a PD control [13] in the translation and rotation axis. Proportional-Derivative (PD) controls are widely used for high-level control in robotics [32, 35, 43, 44]. Similarly, we used a PD controller to create a velocity control based on cartesian coordinates given by Unity. The Kortex Library[3] internally computes inertia, Gravity and Coriolis.

Equation 1 and 2 describe the PD control based on Dorf and Bishop [13]:

$$P_{x,y,z} = P_{x,y,z,bias} + K_p * e(t) + K_d * (e(t) - e_{-1}(t)) \quad (1)$$

$$R_{\theta,\psi,\phi} = R_{\theta,\psi,\phi,bias} + K_{pR} * e(t) + K_{dR} * (e(t) - e_{-1}(t)) \quad (2)$$

Where $P_{x,y,z}$ and $R_{\theta,\psi,\phi}$ represent the pose of the end effector (Cartesian position and rotations), $P_{x,y,z,bias}$ and $R_{\theta,\psi,\phi,bias}$ are the standard offsets in position and rotation, K_p, K_d and K_{pR}, K_{dR} are the controller components (PD) for position and rotation respectively and finally $e(t)$ represent the error between target position/rotation and current position/rotation. We used them to interface the target position and rotation vectors and the robot end-effector position/rotations (joint $q_{[n]}$). Rotations and translations are handled separately, given the mechanical and interaction implications of each of them. While an overshoot in the robot's translation can easily lead to a collision with the user, an overshoot in rotation (in-place) of the end-effector will preserve the distance between the end-effector to the user's hand. The velocities (Cartesian and angular) are introduced in the Gaussian Damped Least Square inverse kinematics solver. This solver is based on Jacobian inversion and adds a gaussian damping factor to handle the behavior of the jacobian matrix near singularity configurations. The reader can find further information about this algorithm in Phuoc et al. [33]. Figure 3 illustrates this control loop in more detail.

The plugin sends cartesian ($\vec{v}$) and angular velocities ($\vec{\omega}$) and reads the robot's joint rotation angles to animate the 3D model in the scene.

3.3 Interface

Our plugin visual interface is divided into two main components: (1) the editor interface that works as GUI input to the robot and

[3]https://github.com/Kinovarobotics/kortex

(2) the 3D representation of the robot that provides feedback about the robot's current pose.

3.3.1 Editor interface. The interface of our plugin (see Figure 2b) enables the user to set *communication parameters* such as IP address and login information (required to access the manipulator), as well as *Target inputs:* which are the goal coordinates and angles that the user wants the end effector to adopt, the pose is communicated in real-time to the robot. Access to these variables is also possible from external scripts by making a call to the script instance. *Control Parameters* are the inputs fields for K_p, K_d, K_{pR}, K_{dR}, these values can be used for online tuning of the robot behavior or for damping the speeds. The field *Measured Position* provides feedback of the end-effector position in the robot base coordinates frame and is drawn using a gizmo in the Graphic engine 3d space.

3.3.2 Robot 3D representation. We render the robot's 3D model in the virtual environment using the values obtained from the cobot's encoders. Herefore, we use the values of the forward kinematics that are implicitly calculated by setting those rotations to the 3d subcomponents of the cobot (*gameobjects*). We also render the forward kinematics obtained from the Kortex solver as a gizmo in the debug window of the 3D engine. Our library assumes that the robot is in the center of the coordinate system, as this is not always the case; the control script of the robot harmonizes the mismatch in coordinate systems using the virtual robot 3D position. Therefore, the end-effector can be set to follow an object in the scene without requiring to provide relative coordinates of the robot, allowing a more straightforward game logic.

4 TECHNICAL EVALUATION

To test the functionality and stability of our plugin, we ran a set of technical tests and showcased the capabilities of the system by developing two common use-cases in haptics for VR; encountered-type haptics, and object manipulation.

Our system has two primary design criteria: (1) Safety of use in shared environments with humans, and (2) speed performance to timely meet the user requirements. To facilitate the understanding of the dynamics of the system, we characterized the response of the cobot using Cobity, detailed values about the transfer function can be found in the plugin repository.

We executed an automated test of the robot speed in every translational axis. The robot was programmed to move from position A to B ($A - B = 40cm$). Then, we recorded the time it needed to reach the final position. Similarly, we evaluated the system's rotation axis; We rotated the end effector from an angle α_a to α_b ($\alpha_a - \alpha_b = 90°$). Figure 4 shows the step responses for the system without any PD control tuning. The standard response of the system is constrained in speed in order to meet the regulations for cobots (EN ISO 10218-2:2011).

5 SHOWCASES

In the following, we present four showcases that highlight the potential for rapid prototyping of our plug-and-play solution. For each application, we connect the virtual environment with the cobot to deliver haptic feedback.

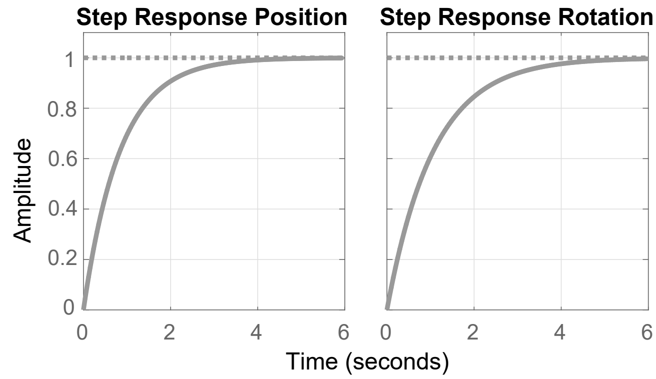


Figure 4: Step Responses to Rotational and translational degrees of freedom (Normalized response)

Encountered-type Haptics. When using a robotic manipulator, encountered-type haptics (ETH) is probably one of the first use cases to imagine. We implemented a virtual environment to replicate a simple ETH scenario; the participant's hand can be tracked using VIVE Trackers or Leap motion (or the headset built-in hand tracking). A straightforward approach to enable ETH is to move the robot in the plane of the surface that is required to be rendered, constrained to the hand's movement. Using our plugin, it is necessary only to trace a line from the hand position to the surface to be rendered and move the robot according to the hand movement. Further improvements can be added, for example making a predictive control of the end effector to anticipate the hand's future position. Figure 5a and 5b depicts a user touching a flat surface that corresponds to a virtual door. A more realistic rendering of flat surfaces can be obtained using a round rotating surface as described in [25]. The latter approach also enables the rendering of different surface textures.

Mid-air Extended Haptics. Ultrasound-based mid-air haptics uses sound waves generated by an array of transducers to render tactile sensations at the palm of the hand. A well-known constraint of mid-air haptics is the rendering workspace [2, 8, 17, 39], the usage of a cobot as a driver of the haptic array can help to overcome such limitations. State of the art approaches proposes to increase the number of arrays [39], attach the array to a rotary joint [17] or switch the positions of the array as required [8].

However, an online driving of the array in the 3D space using a manipulator emerges as a more robust and beneficial approach. We attached a haptic array to the end effector of our robot and guided the robot's movement using the position of the palm, given by a Leap Motion. Using the transform of the end effector (simulated thanks to the rotation of the joints read by the plugin), we transformed the coordinates of the hand from the leap motion coordinates to the world coordinates. Figure 5c and 5d illustrate a user interacting with a dynamic haptic array; the array is kept at a distance of ~30cm of the palm to preserve a high rendering quality.

Social Touch. Social VR has been increasing its presence in VR stores; apps like VR chat, RecRoom or PokerStars VR are becoming more popular. Social VR allows multiple users to join in a shared

(a) Encoutered Type Haptics (b) Encoutered Type Haptics Virtual Environment (c) Extended Mid-Air Haptics (d) Extended Mid-Air Haptics Virtual Environment

Figure 5: Four additional example use cases which we can envision to deliver haptic feedback in VR.

Virtual Environment and let the participants interact in a more natural way than 2D interactions. However, touch is still a missing component in this context. Social touch has been demonstrated to increase the perceived human likeliness in virtual agents [15]. However, this sensation depends highly on the kinaesthetic feedback provided by the human hand; therefore, although versatile, vibrotactile actuation does not create such perception. Alternatively, robotic manipulators with human hand alike end effectors can automate this task and enhance social VR environments.

Figure 1 shows our implementation of social touch using a silicon human hand that features a heat-able foil. The prototype is driven by a cobot using the Cobity plugin. The participant interacts in a VR environment. Whenever the virtual avatar touches the user's shoulder, the robot moves the hand to their real shoulder. This setup could be further improved by using rigged hands as end effectors, for example, the Shadow Dexterous Hand[4].

On Demand Tangible objects. Tangible props are a common approach to introducing haptics in a VR scene, yet, the usage of Ttngibles requires a previous preparation of the physical scene to match the haptic-enabled VR objects, reducing the flexibility of this method. Recent advances have demonstrated that it is possible to alter the perception of Stiffness and friction of tangible prop [12, 38]. Furthermore, De Tynguy [11] explored the extent to which the virtual representation of tangible props can be altered without perceiving such mismatch. In addition to those approaches, the ability to switch the tangibles presented on the scene would significantly enhance the versatility of such a method. Mercado et al. [24] presented a remarkable proof of concept addressing this use case.

6 DISCUSSION AND CONCLUSION

Cobot control remains a highly technical task, keeping apart HCI designers with non-engineering backgrounds. Current solutions require deploying extensive middle-ware and, in some cases, involve more than one operative system for simple prototyping. To facilitate introducing cobots in VR applications and reduce the time of experience prototyping, we introduced Cobity, a solution to use a virtual environment to control the cobot directly from Unity. We developed a real-time C++ library acting as a bridge between the cobot and the graphic engine. In this way, we transfer the 3D positions of the VR application onto real cobot motions.

We presented a dynamic library that enables the usage of Unity as rapid cobot experience prototyping. We run a technical evaluation using our new plugin. Finally, we presented a range of showcases that evidence the flexibility of cobot-based haptics, from kinaesthetic to ethereal sensations, including social touch. The goal of Cobity is to facilitate rapid prototyping for cobot usage in VR instead of replacing the standard architecture ROS. The purpose of such a plugin is to speed up the prototyping of applications within the HCI domain. In the bigger picture, this plugin will facilitate the mediation with different types of cobots used for HCI research.

> Cobity is available at https://github.com/xteeven/Cobity and maintained by the Media Informatics Group at LMU Munich. New features and development will be added to this repository.

As of today, our system only considers the serial robot Kinova Gen3 (6DoF + 7Dof) and Gen3 Lite; we envision a compatibility enlargement to include other widely-used models as the Universal Robots line (UR3, UR5, and UR10), as this line of cobots is more common in HCI environments. The following stages of the plugin require implementing a simulation system to help designers have development speed even higher. Moreover, individual joint control is required for more complex scenarios, allowing a more comprehensive range of applications for our plugin, especially those that demand path planning or obstacle avoidance.

ACKNOWLEDGMENTS

This project was funded by the European Research Council (ERC AMPLIFY, no. 683008) and the Federal Ministry of Education and Research of Germany (BMBF Hive, no. 16SV8183).

REFERENCES

[1] Víctor Hugo Andaluz, Fernando A. Chicaiza, Cristian Gallardo, Washington X. Quevedo, José Varela, Jorge S. Sánchez, and Oscar Arteaga. 2016. Unity3D-MatLab Simulator in Real Time for Robotics Applications. In *Augmented Reality, Virtual Reality, and Computer Graphics*, Lucio Tommaso De Paolis and Antonio Mongelli (Eds.). Springer International Publishing, Cham, 246–263. https://doi.org/10.1007/978-3-319-40621-3_19

[4] https://www.shadowrobot.com/dexterous-hand-series/

[2] Kentaro Ariga, Masahiro Fujiwara, Yasutoshi Makino, and Hiroyuki Shinoda. 2021. Workspace Evaluation of Long-Distance Midair Haptic Display Using Curved Reflector. In *2021 IEEE World Haptics Conference (WHC)*. IEEE, Virtual, 85–90. https://doi.org/10.1109/WHC49131.2021.9517193

[3] Edwin Babaians, Mohsen Tamiz, Yaser Sarfi, Amir Mogoei, and Esmaeil Mehrabi. 2018. ROS2Unity3D; High-Performance Plugin to Interface ROS with Unity3d engine. In *2018 9th Conference on Artificial Intelligence and Robotics and 2nd Asia-Pacific International Symposium*. IEEE, Iran, 59–64. https://doi.org/10.1109/AIAR.2018.8769798

[4] Christoph Bartneck, Marius Soucy, Kevin Fleuret, and Eduardo B. Sandoval. 2015. The robot engine — Making the unity 3D game engine work for HRI. In *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, Kobe, Japan, 431–437. https://doi.org/10.1109/ROMAN.2015.7333561

[5] Andrea Bisson. 2020. COMAU: Collaborative Robotics Market and Applications in Industrial Environments. In *Inclusive Robotics for a Better Society*, José L. Pons (Ed.). Springer International Publishing, Cham, 12–16. https://doi.org/10.1007/978-3-030-24074-5_3

[6] Mario Bollini, Stefanie Tellex, Tyler Thompson, Nicholas Roy, and Daniela Rus. 2013. *Interpreting and Executing Recipes with a Cooking Robot.* Springer International Publishing, Heidelberg, 481–495. https://doi.org/10.1007/978-3-319-00065-7_33

[7] Elodie Bouzbib, Gilles Bailly, Sinan Haliyo, and Pascal Frey. 2021. Can I Touch This?: Survey of Virtual Reality Interactions via Haptic Solutions, In 32e Conférence Francophone sur l'Interaction Homme-Machine (IHM '20.21). *ArXiv* abs/2101.11278, 1–11. arXiv:2101.11278 [cs.HC]

[8] Daniel Brice, Thomas McRoberts, and Karen Rafferty. 2019. A Proof of Concept Integrated Multi-systems Approach for Large Scale Tactile Feedback in VR. In *Augmented Reality, Virtual Reality, and Computer Graphics*, Lucio Tommaso De Paolis and Patrick Bourdot (Eds.). Springer International Publishing, Cham, 120–137. https://doi.org/10.1007/978-3-030-25965-5_10

[9] Christopher Crick, Graylin Jay, Sarah Osentoski, Benjamin Pitzer, and Odest Chadwicke Jenkins. 2017. *Rosbridge: ROS for Non-ROS Users.* Springer International Publishing, Cham, 493–504. https://doi.org/10.1007/978-3-319-29363-9_28

[10] Edson de Araújo Silva, Eduardo Valentin, Jose Reginaldo Hughes Carvalho, and Raimundo da Silva Barreto. 2021. A survey of Model Driven Engineering in robotics. *Journal of Computer Languages* 62 (2021), 101021. https://doi.org/10.1016/j.cola.2020.101021

[11] Xavier de Tinguy, Claudio Pacchierotti, Mathieu Emily, Mathilde Chevalier, Aurélie Guignardat, Morgan Guillaudeux, Chloé Six, Anatole Lécuyer, and Maud Marchal. 2019. How Different Tangible and Virtual Objects Can Be While Still Feeling the Same?. In *2019 IEEE World Haptics Conference (WHC)*. IEEE, Tokyo, Japan, 580–585. https://doi.org/10.1109/WHC.2019.8816164

[12] Xavier de Tinguy, Claudio Pacchierotti, Maud Marchal, and Anatole Lécuyer. 2018. Enhancing the Stiffness Perception of Tangible Objects in Mixed Reality Using Wearable Haptics. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, Tuebingen, Germany, 81–90. https://doi.org/10.1109/VR.2018.8446280

[13] Richard C Dorf and Robert H Bishop. 2011. *Modern control systems.* Pearson, Singapore.

[14] Sebastian Günther, Florian Müller, Felix Hübner, Max Mühlhäuser, and Andrii Matviienko. 2021. ActuBoard: An Open Rapid Prototyping Platform to Integrate Hardware Actuators in Remote Applications. In *Companion of the 2021 ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (Virtual Event, Netherlands) *(EICS '21)*. Association for Computing Machinery, New York, NY, USA, 70–76. https://doi.org/10.1145/3459926.3464757

[15] Matthias Hoppe, Beat Rossmy, Daniel Peter Neumann, Stephan Streuber, Albrecht Schmidt, and Tonja-Katrin Machulla. 2020. A Human Touch: Social Touch Increases the Perceived Human-Likeness of Agents in Virtual Reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3313831.3376719

[16] Leire Amezua Hormaza, Wael M Mohammed, Borja Ramis Ferrer, Ronal Bejarano, and Jose L Martinez Lastra. 2019. On-line training and monitoring of robot tasks through virtual reality. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, Vol. 1. IEEE, Finland, 841–846. https://doi.org/10.1109/INDIN41052.2019.8971967

[17] Thomas Howard, Maud Marchal, Anatole Lécuyer, and Claudio Pacchierotti. 2020. PUMAH: Pan-Tilt Ultrasound Mid-Air Haptics for Larger Interaction Workspace in Virtual Reality. *IEEE Transactions on Haptics* 13, 1 (2020), 38–44. https://doi.org/10.1109/TOH.2019.2963028

[18] Lentin Joseph. 2018. *Robot Operating System (ROS) for Absolute Beginners.* Springer, Singapore. https://doi.org/10.1007/978-1-4842-3405-1

[19] Dahyun Kang, Sonya S. Kwak, Hanbyeol Lee, and JongSuk Choi. 2020. First Things First: A Survey Exploring Key Services and Functions of a Robot. In *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction* (Cambridge, United Kingdom) *(HRI '20)*. Association for Computing Machinery, New York, NY, USA, 278–280. https://doi.org/10.1145/3371382.3378317

[20] Ben Kehoe, Sachin Patil, Pieter Abbeel, and Ken Goldberg. 2015. A Survey of Research on Cloud Robotics and Automation. *IEEE Transactions on Automation Science and Engineering* 12, 2 (2015), 398–409. https://doi.org/10.1109/TASE.2014.2376492

[21] Yaesol Kim, Hyun Jung Kim, and Young J. Kim. 2018. Encountered-type haptic display for large VR environment using per-plane reachability maps. *Computer Animation and Virtual Worlds* 29, 3-4 (2018), e1814. https://doi.org/10.1002/cav.1814 e1814 cav.1814.

[22] Xin Li, José-Fernán Martínez, Jesús Rodríguez-Molina, and Néstor Lucas Martínez. 2016. A Survey on Intermediation Architectures for Underwater Robotics. *Sensors* 16, 2 (2016), 0–0. https://doi.org/10.3390/s16020190

[23] Ali Ahmad Malik, Tariq Masood, and Arne Bilberg. 2020. Virtual reality in manufacturing: immersive and collaborative artificial-reality in design of human-robot workspace. *International Journal of Computer Integrated Manufacturing* 33, 1 (2020), 22–37. https://doi.org/10.1080/0951192X.2019.1690685

[24] Victor Rodrigo Mercado, Thomas Howard, Hakim Si-Mohammed, Ferran Argelaguet, and Anatole Lécuyer. 2021. Alfred: the Haptic Butler On-Demand Tangibles for Object Manipulation in Virtual Reality using an ETHD. In *2021 IEEE World Haptics Conference (WHC)*. IEEE, Tokyo, Japan, 373–378. https://doi.org/10.1109/WHC49131.2021.9517250

[25] Victor Rodrigo Mercado, Maud Marchal, and Anatole Lécuyer. 2021. ENTROPiA: Towards Infinite Surface Haptic Displays in Virtual Reality Using Encountered-Type Rotating Props. *IEEE Transactions on Visualization and Computer Graphics* 27, 3 (2021), 2237–2243. https://doi.org/10.1109/TVCG.2019.2963190

[26] Víctor Rodrigo Mercado, Maud Marchal, and Anatole Lécuyer. 2021. "Haptics On-Demand": A Survey on Encountered-Type Haptic Displays. *IEEE Transactions on Haptics* 14, 3 (2021), 449–464. https://doi.org/10.1109/TOH.2021.3061150

[27] Joseph E. Michaelis, Amanda Siebert-Evenstone, David Williamson Shaffer, and Bilge Mutlu. 2020. Collaborative or Simply Uncaged? Understanding Human-Cobot Interactions in Automation. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376547

[28] Carl A Moore, Michael A Peshkin, and J Edward Colgate. 2003. Cobot implementation of virtual paths and 3D virtual surfaces. *IEEE Transactions on Robotics and Automation* 19, 2 (2003), 347–351. https://doi.org/10.1109/TRA.2003.808866

[29] Thomas Muender, Anke V. Reinschluessel, Sean Drewes, Dirk Wenig, Tanja Döring, and Rainer Malaka. 2019. Does It Feel Real? Using Tangibles with Different Fidelities to Build and Explore Scenes in Virtual Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300903

[30] Tomi Nukarinen, Jari Kangas, Jussi Rantala, Toni Pakkanen, and Roope Raisamo. 2018. Hands-Free Vibrotactile Feedback for Object Selection Tasks in Virtual Reality. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology* (Tokyo, Japan) *(VRST '18)*. Association for Computing Machinery, New York, NY, USA, Article 94, 2 pages. https://doi.org/10.1145/3281505.3283375

[31] Carl Oechsner, Sven Mayer, and Andreas Butz. 2022. Challenges and Opportunities of Cooperative Robots as Cooking Appliances. In *Proceedings of the 2022 Workshop on Engaging with Automation* (2022-04-30) *(AutomationXP22)*. CEUR Workshops, Germany, 7 pages. http://ceur-ws.org/Vol-3154/paper14.pdf

[32] Puren R Ouyang, J Tang, WH Yue, and Shanuka Jayasinghe. 2016. Adaptive PD plus sliding mode control for robotic manipulator. In *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, Alberta, Canada, 930–934. https://doi.org/10.1109/AIM.2016.7576888

[33] Le Minh Phuoc, Philippe Martinet, Sukhan Lee, and Hunmo Kim. 2008. Damped least square based genetic algorithm with Ggaussian distribution of damping factor for singularity-robust inverse kinematics. *Journal of mechanical science and technology* 22, 7 (2008), 1330–1338. https://doi.org/10.1007/s12206-008-0427-4

[34] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. 2009. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*. IEEE, Stanford, 0–0.

[35] Marina Ramon Borràs. 2020. *iMOVE: Development of a hybrid control interface based on sEMG and movement signals for an assistive robotic manipulator.* Master's thesis. Universitat Politècnica de Catalunya.

[36] Jose Luis Rubio-Tamayo, Manuel Gertrudix Barrio, and Francisco García García. 2017. Immersive Environments and Virtual Reality: Systematic Review and Advances in Communication, Interaction and Simulation. *Multimodal Technologies and Interaction* 1, 4 (2017), 0–0. https://doi.org/10.3390/mti1040021

[37] Saeed Saeedvand, Masoumeh Jafari, Hadi S. Aghdasi, and Jacky Baltes. 2019. A comprehensive survey on humanoid robot development. *The Knowledge Engineering Review* 34 (2019), e20. https://doi.org/10.1017/S0269888919000158

[38] Steeven Villa Salazar, Claudio Pacchierotti, Xavier de Tinguy, Anderson Maciel, and Maud Marchal. 2020. Altering the Stiffness, Friction, and Shape Perception of Tangible Objects in Virtual Reality Using Wearable Haptics. *IEEE Transactions on Haptics* 13, 1 (2020), 167–174. https://doi.org/10.1109/TOH.2020.2967389

[39] Shun Suzuki, Masahiro Fujiwara, Yasutoshi Makino, and Hiroyuki Shinoda. 2019. Midair Ultrasound Haptic Display with Large Workspace. In *Haptic Interaction*,

Hiroyuki Kajimoto, Dongjun Lee, Sang-Youn Kim, Masashi Konyo, and Ki-Uk Kyung (Eds.). Springer Singapore, Singapore, 3–5.

[40] Unity-Technologies. 2022. Unity-Technologies/Unity-robotics-hub: Central Repository for tools, tutorials, resources, and documentation for robotics simulation in Unity. https://github.com/Unity-Technologies/Unity-Robotics-Hub

[41] Manuela M. Veloso. 2018. The Increasingly Fascinating Opportunity for Human-Robot-AI Interaction: The CoBot Mobile Service Robots. *J. Hum.-Robot Interact.* 7, 1, Article 5 (May 2018), 2 pages. https://doi.org/10.1145/3209541

[42] Dangxiao Wang, Yuan Guo, Shiyi Liu, Yuru Zhang, Weiliang Xu, and Jing Xiao. 2019. Haptic display for virtual reality: progress and challenges. *Virtual Reality and Intelligent Hardware* 1, 2 (2019), 136–162. https://doi.org/10.3724/SP.J.2096-5796.2019.0008

[43] Chifu Yang, Qitao Huang, Hongzhou Jiang, O Ogbobe Peter, and Junwei Han. 2010. PD control with gravity compensation for hydraulic 6-DOF parallel manipulator. *Mechanism and Machine theory* 45, 4 (2010), 666–677. https://doi.org/10.1016/j.mechmachtheory.2009.12.001

[44] Dongya Zhao, Yongrui Zhao, Shaoyuan Li, and Quanmin Zhu. 2011. A new terminal converging PD control for parallel robotic manipulators with bounded torque. In *Proceedings of 2011 International Conference on Modelling, Identification and Control*. IEEE, Innsbruck, Austria, 115–121. https://doi.org/10.1109/ICMIC.2011.5973686