Delft University of Technology

Physics Engine-Based Whole-Hand Haptic Rendering for Sensorimotor Neurorehabilitation

Ratz, Raphael; Marchal-Crespo, Laura

DOI
10.1109/WHC56415.2023.10224404

Publication date
2023

Document Version
Final published version

Published in
Proceedings of the 2023 IEEE World Haptics Conference, WHC 2023

Citation (APA)
Ratz, R., & Marchal-Crespo, L. (2023). Physics Engine-Based Whole-Hand Haptic Rendering for Sensorimotor Neurorehabilitation. In *Proceedings of the 2023 IEEE World Haptics Conference, WHC 2023* (pp. 279-285). IEEE. https://doi.org/10.1109/WHC56415.2023.10224404

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

Otherwise as indicated in the copyright section: the publisher
is the copyright holder of this work and the author uses the
Dutch legislation to make this work public.

2023 IEEE World Haptics Conference (WHC) | 979-8-3503-9993-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/WHC56415.2023.10224404

Physics Engine-Based Whole-Hand Haptic Rendering for Sensorimotor Neurorehabilitation

Raphael Rätz[1] and Laura Marchal-Crespo[1,2,3]

***Abstract*—Whole-hand haptic rendering could lead to more naturalistic and intuitive virtual hand-object interactions, which could be especially beneficial for applications such as sensorimotor robotic neurorehabilitation. However, the majority of previously proposed whole-hand haptic rendering algorithms rely on effortful custom implementations or are not suited for the grounded haptic devices often used in neurorehabilitation. Therefore, we suggest a framework for whole-hand haptic rendering based on a readily available physics engine. We employ a bilateral position-position teleoperation framework between a haptic rehabilitation device and a simulated hand avatar with added exercise-specific haptic rendering. Moreover, in consideration of the needs of neurological patients, we introduce an adaptive damping of the haptic device during hand-object interactions for increased stabilization of the patient's limb. We present the first results of the feasibility of the proposed framework in a haptic rehabilitation exercise. In an ongoing clinical study, the practical application of the presented framework is currently investigated.**

***Index Terms*—Haptics, physics engine, neurorehabilitation, grasping, hand.**

I. INTRODUCTION

Robotic devices have been used in neurorehabilitation research for more than 25 years and have become a promising tool for delivering high-intensity sensorimotor training [1], [2]. Especially in conjunction with the gamification of therapy, robots allow performing highly repetitive training in engaging virtual environments [3]. Although neuroscience highlights the importance of sensory information during training (e.g., [4], [5]), the interactions with virtual game elements are still mostly visually driven or rely on basic haptic effects [6]. To investigate the impact of task-specific high-fidelity haptic interactions during training, adequate robotic devices and haptic rendering frameworks are required. In the realm of upper-limb rehabilitation of patients with sensorimotor deficits, the importance of training hand functions such as grasping is emphasized in literature [2]. Hand-object interactions are therefore of particular interest when it comes to designing virtual training tasks.

For haptic rendering of hand-object interactions, the use of multiple predetermined interaction points in the form of spherical colliders (i.e., virtual representation used to compute collisions) between a hand avatar (i.e., a virtual hand representation) and virtual objects has been proposed [7]. With haptic rendering libraries such as *Chai3D* [8], this can be relatively easy to implement. However, this method is inherently limited to simulating physical interactions only at predefined interaction points, which may introduce visuo-haptic incongruences if a human hand-like visual representation is shown to the patient. This might hamper motor performance [9] and could result in an undesired increase in patients' cognitive load, especially in patients with cognitive impairments. More realistic whole-hand interactions, where the haptic rendering reflects the entire visual hand representation may lead to more natural interactions with virtual objects [10]. This could not only be advantageous for neurorehabilitation but also for teleoperation [11] and virtual reality (VR) applications such as virtual manufacturing training [12].

One of the first whole-hand haptic rendering algorithms was developed by Tzafestas for an exoskeleton glove [10]. Garre and Otaduy presented a method to simulate the interaction of a hand composed of a rigid main body and deformable digits and a virtual environment [13]. However, the utilized haptic device did not display interaction forces on the fingers. More recently, Tong et al. introduced a framework for haptic gloves, based on a sphere-tree model and articulated cone frustums [14]. Yet, their algorithm aims to be used with non-grounded devices (e.g., gloves) and does not consider the physics (e.g., inertial properties) of interactive virtual objects. An et al. presented a method for a two-finger haptic device, but their approach is limited to approximating the dynamics of tangible objects by pendulum models [15]. Lobo et al. presented a proxy-based algorithm for underactuated and non-grounded one degree of freedom (DoF) devices where they computed the force projection on the actuated DoF based on a quadratic optimization [16]. Verschoor et al. presented CLAP, an approach that is capable of simulating hand-object interactions through a stable soft tissue simulation and that provides easy integration in game and physics engines [17]. However, it was developed for VR applications with reported update rates of 60-100 Hz. Despite the authors advocating its use in haptics, to our knowledge, no application in haptic devices has been published to date.

Although the above hand-object rendering approaches exhibit great performance in specific use cases, they often lack flexibility, ease of implementation, the possibility for an interactive dynamic environment, and/or cannot be applied to grounded haptic devices. We, therefore, endorse the use of

---

*This work was supported by the Innosuisse grant 32213.1 IP-CT and the Dutch Research Council (NWO) Talent Program VIDI TTW 2020

[1] Raphael Rätz and Laura Marchal-Crespo are with the Motor Learning and Neurorehabilitation Laboratory, ARTORG Center for Biomedical Engineering Research, University of Bern, Switzerland `raphael.raetz@unibe.ch`

[2] Laura Marchal-Crespo is also with the Department of Cognitive Robotics, Delft University of Technology, Delft, the Netherlands `l.marchalcrespo@tudelf.nl`

[3] Laura Marchal-Crespo is also with the Department Rehabilitation Medicine, Erasmus Medical Center, Rotterdam, the Netherlands

an off-the-shelf physics engine for haptic rendering of whole-hand interactions. While the use of physics engines has been tested in different haptic applications (e.g., [18], [19]), it is often custom-tailored to the specific application, and therefore, lacks the flexibility needed, for example, for the development of virtual therapy tasks with a variety of interactive objects. Several authors have compared different physics engines for robotic applications (e.g., [20]–[22]). It has been noted that even though the *Bullet* physics engine does not seem to be actively further developed, it benefits from a large community and stable contact simulations with high-DoF multi-link bodies. Due to this and the outstanding flexibility in regards to separating visualization and computation, we decided to use *Bullet*.

From a control perspective, physics engine-based haptic rendering can be considered as a bilateral teleoperation system. Hereby, the simulated hand avatar (i.e., teleoperated entity), the controller, and the environment are implemented virtually. For the rest of the paper, we will orient ourselves in the concepts of bilateral teleoperation.

Here, we present our efforts in developing haptic rendering of naturalistic whole-hand haptic interactions in the context of a virtual neurorehabilitation exercise. We leverage an off-the-shelf physics engine and combine it with a classical virtual wall to generate different haptic sensations depending on the exercise task. The proposed algorithm is implemented on a haptic device with three translational DoF for the hand base and three DoF for the fingers and thumb. The haptic rendering is integrated in a neurorehabilitation game that consists of grasping dispensers containing liquids of different viscosities and filling glasses without spilling any liquid [23].

II. METHODS

A. The Haptic Main Device

The haptic main device used in this work (Fig. 1) was specifically developed for sensorimotor neurorehabilitation. A haptic delta robot (a modified Lambda with three DoF, Force Dimension, Switzerland), enables translational movements (x_m, y_m, z_m) of the patient's hand [24]. On top of this, the PRIDE hand module with finger and thumb actuation allows performing grasping movements [25]. PRIDE allows collective flexion/extension of all fingers (represented by the fingertip orientation $\theta_{f,m}$), thumb circumduction ($\theta_{c,m}$), and thumb flexion/extension ($\theta_{t,m}$). The thumb flexion/extension mechanism is in series with the circumduction joint (see [26] for further details). Thus, the 6×1 vector of generalized main device coordinates $\mathbf{q}_m$ is characterised as $\mathbf{q}_m = [x_m, y_m, z_m, \theta_{f,m}, \theta_{c,m}, \theta_{t,m}]^T$. The main device forces/torques are described by the vector $\mathbf{f}_m = [f_{x,m}, f_{y,m}, f_{z,m}, f_{f,m}, \tau_{c,m}, f_{t,m}]^T$. Albeit we use angular finger tip and thumb tip positions $\theta_{f,m}$ and $\theta_{t,m}$, we describe the finger and thumb flexion/extension efforts with the forces $f_{f,m}$ and $f_{t,m}$ for easier interpretability. See [25] and [26] for details and for the conversion to motor torques.

B. The Simulated Whole Hand Avatar

The simulated hand avatar was modelled in *Bullet* with capsule colliders. The corresponding vector of generalized coordinates $\mathbf{q}_s$ and the force vector $\mathbf{f}_s$ are analogous to the haptic main device (Fig. 2). The finger tip and thumb tip angular positions for flexion/extension are computed as $\theta_{f,s} = \theta_{f,1} + \theta_{f,2} + \theta_{f,3}$ and $\theta_{t,s} = \theta_{t,1} + \theta_{t,2} + \theta_{t,3}$. Each three-link anatomical finger is reduced to one DoF by coupling the three interphalangeal joints. In *Bullet*, this was achieved by resetting the joint positions $\theta_{f,3}$ and $\theta_{f,2}$ to the desired value in each simulation step. Moreover, all fingers are coupled ($\theta_{f,1} = \theta'_{f,1} = \theta''_{f,1} = \theta'''_{f,1}$) using gear constraints, effectively reducing all DoF of the fingers to one collective DoF. Again, finger tip and thumb tip forces $f_{f,s}$ and $f_{f,s}$ will be used. Note that in the simulation, they were converted to joint torques at the most proximal joints (see [25] for the mapping).

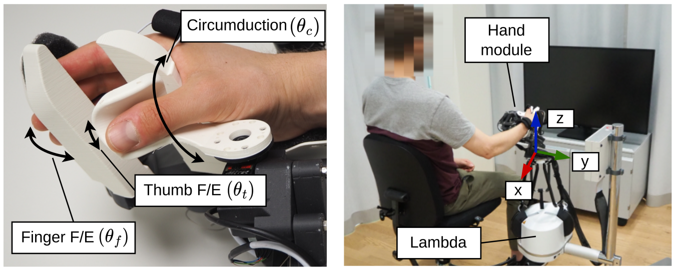

Fig. 1. The used haptic device consists of a Lambda robot and a hand module.

C. Bilateral Teleoperation for Whole-Hand Haptic Rendering

The dynamics of a multi-link teleoperated entity such as the simulated hand avatar can be represented as follows:

$$\mathbf{M}(\mathbf{q}_s)\ddot{\mathbf{q}}_s[k] + \mathbf{h}(\dot{\mathbf{q}}_s[k], \mathbf{q}_s[k]) = \mathbf{f}_s[k], \tag{1}$$

where the term $\mathbf{M}(\mathbf{q}_s)$ denotes the mass matrix, $\mathbf{h}(\dot{\mathbf{q}}_s, \mathbf{q}_s)$ is the lumped term for velocity and gravity dependent forces, and $\mathbf{f}_s$ are applied forces such as controller inputs. From here, we will omit the notation of the time step $[k]$ and the dependency of $\mathbf{M}$ and $\mathbf{h}$ from position and velocity respectively for the sake of readability. In the rest of this section, it can be assumed that all expressions are evaluated at time step $[k]$.

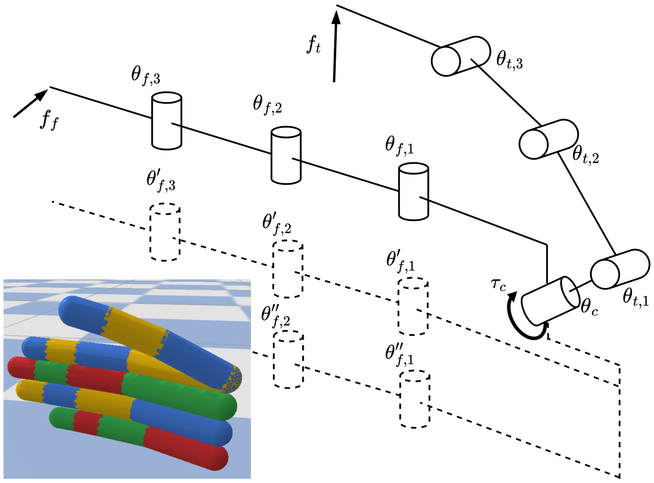

Fig. 2. The kinematics of the simulated whole hand avatar.

We chose to use a position-position control scheme, which is a two-channel variant of the general teleoperation scheme presented by Lawrence [27], corresponding to a modified bilateral proportional-derivative (PD) control. The main device velocity is computed by Euler-backwards differentiation with consecutive low-pass filtering for noise reduction, denoted by $\dot{\mathbf{q}}_{m,f}$. Here, a first-order low-pass filter discretised by an Euler-backwards scheme was used. The vectors of main device ($\mathbf{f}_m$) and simulated hand avatar ($\mathbf{f}_s$) forces are generated through the control blocks $\mathbf{C}_{mm}$, $\mathbf{C}_{ms}$, $\mathbf{C}_{ss}$, $\mathbf{C}_{sm}$ and $\mathbf{C}_{ma}$ (see Fig. 3 for the control architecture) and are provided in Eq. (2) and Eq. (3), using the diagonal proportional gain matrices $\mathbf{K}_{p,m}$, $\mathbf{K}_{p,s}$ and damping gain matrices $\mathbf{K}_{d,m}$, $\mathbf{K}_{d,s}$.

$$\mathbf{f}_s = \tilde{\mathbf{M}} \left(\mathbf{K}_{p,s}(\mathbf{q}_m - \mathbf{q}_s) + \mathbf{K}_{d,s}(\dot{\mathbf{q}}_{m,f} - \dot{\mathbf{q}}_s) \right) + \mathbf{h} \tag{2}$$

$$\mathbf{f}_m = \mathbf{K}_{p,m}(\mathbf{q}_s - \mathbf{q}_m) - \boldsymbol{\Phi}\mathbf{K}_{d,m}\dot{\mathbf{q}}_{m,f} + \mathbf{f}_e \tag{3}$$

The term $\mathbf{h}$ in Eq. 2 is obtained from an inverse dynamics computation with zero accelerations and is provided by the physics engine and the term $\tilde{\mathbf{M}}$ denotes the mass matrix only containing the diagonal elements and can also be obtained from the physics engine. The diagonal matrix $\boldsymbol{\Phi}$ in Eq. 3 is obtained from an adaptive control law and $\mathbf{f}_e$ is a vector of additional exercise-specific forces (see sections II-D and II-F). Substituting $\mathbf{f}_s$ in Eq. (1) with Eq. (2), we obtain the simulated hand avatar acceleration in free space:

$$\ddot{\mathbf{q}}_s = \mathbf{M}^{-1}\tilde{\mathbf{M}} \left(\mathbf{K}_{p,s}(\mathbf{q}_m - \mathbf{q}_s) + \mathbf{K}_{d,s}(\dot{\mathbf{q}}_{m,f} - \dot{\mathbf{q}}_s) \right). \tag{4}$$

To linearize the simulated hand avatar dynamics –i.e., taking into account the position-dependent changes of inertial characteristics (e.g., the finger inertia decreases during flexion)– we multiply the simulated hand avatar PD controller output by $\tilde{\mathbf{M}}$. It could be tempting to use $\tilde{\mathbf{M}} = \mathbf{M}$ which would effectively decouple the system in free space. This is also known as computed torque controller [28]. However, it is not suitable in our application because the off-diagonal elements of $\tilde{\mathbf{M}} = \mathbf{M}$ would result in parasitic torques/forces (i.e., forces/torques that are undesirably reflected onto other DoF) during hand-object interactions.

D. Adaptive Damping

The control block $\mathbf{C}_{ma}$ in Fig. 3 describes an adaptive damping that aims to stabilize the patient's hand, fingers and/or thumb upon impact with an object. As neurological patients might suffer from spasticity (i.e., an involuntary and undesired reaction to abrupt movements of their limbs), we wanted to implement a mechanism of stabilization that allows feeling hand-object impacts, but that would only minimally influence slow movements in free space. Let $\boldsymbol{\Phi}(t)$ be a time-variant diagonal matrix where each diagonal element takes a value in the interval $[0, 1]$. To compute this matrix, first, the activation functions for the finger, thumb circumduction and thumb flexion/extension (i.e., ϕ_f, ϕ_c, ϕ_t) are updated as stated in Algorithm 1. Hereby, T is the update period and f_0 is a force (or torque for thumb circumduction) threshold at which full activation ($\phi = 1$) is desired. The idea of the algorithm is to quickly activate (i.e., increase the output of the activation function) in case of force/torque spikes experienced by the patient and then gradually fade out with an exponential decay. First, a normalized force value $\rho \in [0, 1]$ based on the magnitude of the respective joint force/torque (denoted as f in Algorithm 1) and normalized over a predefined f_0 is computed. If this value is higher than the output of the activation function ϕ from the previous update of the control loop, the exponential decay function will be reset (by setting $t = 0$) and the variable that stores the peak value of the normalized force magnitude $\hat{\rho}$ is set to the current ρ. In the following updates, the function will decay with a time constant τ_{dec}. Finally, the output of the exponential decay is multiplied by the normalized force peak $\hat{\rho}$ to scale the output according to the last force peak. The diagonal matrix $\boldsymbol{\Phi}$ is then computed in Eq. (5). The maximum value across finger, thumb circumduction and thumb flexion/extension ($\hat{\phi}$) is used for the x-, y- and z-axis. The rationale for this is that we assumed that high stabilization

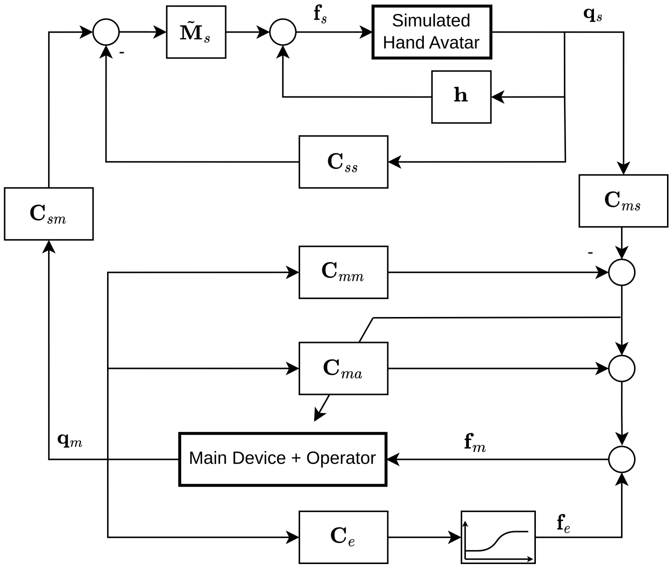

Fig. 3. Bilateral teleoperation control, based on the two-channel position-position control scheme as presented in [27]. Note the addition of the assistive controller $\mathbf{C}_{ma}$, the exercise-specific controller $\mathbf{C}_e$ and the transition block.

Algorithm 1 Computation of the adaptive damping ϕ

```
initialize:
$\hat{\rho} \leftarrow 0,\ t \leftarrow 0,\ \phi \leftarrow 0$
while haptic rendering is running, each update do
    $\rho \leftarrow \min(\frac{|f|}{f_0}, 1)$
    if $\rho \geq \phi$ then
        $t \leftarrow 0$
        $\hat{\rho} \leftarrow \rho$
    else
        $t \leftarrow t + T$
    end if
    $\phi \leftarrow \exp(-\frac{t}{\tau_{dec}})\hat{\rho}$
end while
```

through the adaptive damping would be beneficial in case of any collisions with the hand (i.e., hand base, fingers or thumb). Fig. 7 shows an example of how this activation function reacts to bumps of different magnitudes.

$$\mathbf{\Phi} = \operatorname{diag}\left(\hat{\phi}, \hat{\phi}, \hat{\phi}, \phi_f, \phi_c, \phi_t\right), \quad \hat{\phi} = \max(\phi_f, \phi_c, \phi_t) \tag{5}$$

E. Tuning the Control Gains: Critical Damping for Symplectic Euler Integration

To facilitate the tuning of the hand avatar control gains, we suggest choosing $\mathbf{K}_{d,s}$ in Eq. (2) such that each joint is critically damped. To do this, we need to consider the integration scheme used in *Bullet*. The symplectic Euler integration in Eq. (6) and Eq. (7) –proposed initially by Niiranen [29]– is the de facto standard integration scheme for physics engines [30]–[32] and is also utilized in *Bullet*. The variable T denotes the time increment, k denotes the simulation step, and x the position. Using the time-shifting property of the Z-transform, we obtain the equations in the Z-domain (right side Eq. (6) & (7)).

$$\dot{x}[k{+}1] = \dot{x}[k] + \ddot{x}[k]T \quad \rightarrow \quad \dot{X} = \frac{T}{z-1}\ddot{X} \tag{6}$$

$$x[k{+}1] = x[k] + \dot{x}[k{+}1]T \quad \rightarrow \quad X = \frac{zT}{z-1}\dot{X} \tag{7}$$

To investigate the dynamic behaviour of bodies simulated by symplectic Euler integration, a one-DoF inertia with mass m is considered. If all damping in the physics engine is disabled, its behaviour in free space, subject to an external force f at instant k in discrete time, is as follows:

$$m\ddot{x}[k] = f[k]. \tag{8}$$

Applying Eq. (6) and (7) on Eq. (8), results in the plant transfer function $P(z)$ in Z domain:

$$P(z) = \frac{X(z)}{F(z)} = \frac{1}{m}\frac{zT^2}{(z-1)^2}. \tag{9}$$

We can then apply a PD control law with stiffness and damping gains k_p and k_d respectively, to cancel out the known inertia m as:

$$f_{pd}[k] = m\left(k_p x[k] + k_d \dot{x}[k]\right). \tag{10}$$

Using Eq. (7), this can also be expressed in the Z-domain:

$$C(z) = \frac{F_{pd}(z)}{X(z)} = \frac{m\left(k_d(z-1) + k_p T z\right)}{Tz}. \tag{11}$$

The resulting closed-loop transfer function becomes:

$$G(z) = \frac{CP}{1+CP} = \frac{T^2 k_p z + T k_d (z-1)}{T^2 k_p z + T k_d (z-1) + (z-1)^2}. \tag{12}$$

Hence, the poles of $G(z)$ are:

$$\begin{aligned} z_{1,2} &= \frac{2 - T^2 k_p - T k_d \pm \sqrt{D}}{2} \\ \text{with} \quad D &= T^4 + 2T^3 k_p k_d + T^2(k_d^2 - 4k_p). \end{aligned} \tag{13}$$

We can now impose the two closed-loop poles to be real and coincident in order to obtain a critically damped response. This can be achieved by setting $D = 0$ and solving for k_d to obtain the critical damping gain $k_{d,crit}$ in Eq. (14) (only the positive solution is considered). The gains of the diagonal matrix $\mathbf{K}_{d,s}$ (see section II-C) were computed accordingly for each axis.

$$k_{d,crit} = 2\sqrt{k_p} - k_p T \tag{14}$$

F. Additional Exercise-Specific Haptics

In addition to the whole-hand haptic rendering, we also introduce task-specific haptics whenever a liquid dispenser is squeezed in the cocktailbar game. This is represented by the control block $\mathbf{C}_e$ in Fig. 3. We can make the haptic sensations amongst each of the liquid dispensers more diversified by adding a virtual wall at the fingertips a few mm from each dispenser before the expected contact with the dispenser. By adding this virtual wall, rather than varying the main device control gains to simulate various interactions, we have more flexibility in the design of exercise-specific haptics while still providing the accustomed hand-object interactions. The forces from the whole-hand rendering and the virtual wall are superimposed as soon as the liquid dispenser is touched by the fingers. The additional finger exercise force $f_{f,e}$ from the virtual wall is computed according to Eq. (15) with gains $k_{p,e}$ and $k_{d,e}$, which are adjusted depending on the desired object interaction (i.e., representing liquid dispensers of different viscosities and stiffnesses).

$$f_{f,e} = \begin{cases} k_{p,e}\Delta\theta_{f,m} + k_{d,e}\dot{\theta}_{f,m}, & \Delta\theta_{f,m} > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

To avoid feeling any abrupt appearance of the exercise-specific virtual wall forces, we use a time-dependent transition function $\alpha(t)$. The transition is triggered when the distance between the palm of the hand avatar and one of the liquid dispensers is below a predetermined threshold distance. This distance is computed using an invisible and kinematic body with disabled collisions that is coupled to the hand avatar. The transition is again reversed when this distance exceeds the threshold. A cubic polynomial (see Eq. (16)) is used, where T_{tr} is the transition period and t_{on} denotes the start of the transition. The reverse transition is performed analogously.

$$\alpha(t) = \begin{cases} 0, & t \leq t_{on} \\ \frac{2}{T_{tr}^2}t^2 - \frac{3}{T_{tr}^3}t^3, & t_{on} < t < t_{on} + T_{tr} \\ 1, & t \geq T_{tr} + t_{on} \end{cases} \tag{16}$$

The final additional exercise-specific forces $\mathbf{f}_e$ applied to the device are computed as:

$$\mathbf{f}_e = \alpha[0, 0, 0, f_{f,e}, 0, 0]^T \tag{17}$$

G. Software Architecture

The implemented software architecture is depicted in Fig. 4. The haptic rendering loop updates at 1 kHz and communicates to the haptic device via a custom hardware abstraction layer (HAL). The shared memory option of the C++ API of *Bullet* was used to communicate with a *Bullet* server. This allowed

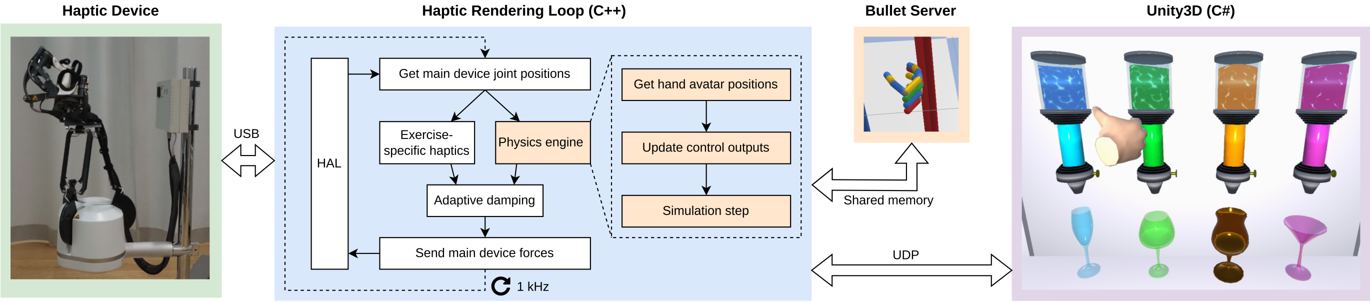

Fig. 4. Software architecture: The device communicates through a hardware abstraction layer (HAL) with the haptic rendering loop. The physics engine is integral part of the 1 kHz haptic loop. For this, a shared memory server of *Bullet* is employed. The communication with the rehabilitation task in *Unity3D* is established through UDP.

us to optionally enable the visualization of the *Bullet* scene. The communication between the neurorehabilitation game in *Unity3D* (implemented in C#) and the haptic rendering loop (implemented in C++) is based on a user datagram protocol (UDP) communication. Only the generalized coordinates of the hand avatar are regularly sent to the game in *Unity3D* – other information is updated asynchronously. For all reported results, the software was executed on a machine with AMD Ryzen 7 5850U CPU, 32 GB RAM and Ubuntu 22.04, 5.15.0 low-latency kernel.

III. RESULTS & DISCUSSION

A. Framework Performance & Control

In Fig. 5, we demonstrate the capability of our approach. Objects with four different shapes, i.e., cube, sphere, cylinder and disk, were placed free-floating in space and then grasped and moved around with the haptic device. The graphical user interface of *Bullet* was used for visualization. We ascertained that complex grasps such as pinch grasps or precision grasps can be effectively performed. To support our claim that the physics engine can be used for real-time haptic rendering in the context of whole-hand interactions, we recorded the execution time of the control loop during 10^5 iterations while actively grasping a cylinder from the neurorehabilitation game and continuously moving all joints. Fig. 6 presents the distribution on a semi-logarithmic scale. The desired 1 ms execution time was met with very few exceptions, thus achieving similar performance as Tong et al. [14].

When it comes to the control and implementation, there are currently shortcomings. The stability of the employed control framework was not examined in this work and the gains were tuned by trial and error. A thorough analysis of the system could lead to conditions for stability and would also allow characterizing the actual impedance perceived by the user, similar to [33]. Moreover, despite reaching an update frequency of 1 kHz in the current application, it is likely that our approach would be limited in more complex scenarios with a larger number of objects or objects with high geometrical complexity. Nevertheless, neurorehabilitation exercises usually only incorporate a limited number of virtual objects to prevent patients from losing their attention to the task at hand. Another inherent limitation of our approach that should be addressed in the future is the slight mismatch between the collider hand based on capsules in *Bullet* and the visual hand avatar rendered in Unity based on a mesh.

B. Final Application: Rehabilitation Exercise

The final application in conjunction with the rehabilitation game is depicted in Fig. 7. In practice, only the game visuals (upper row in Fig. 7) would be shown to the patient during the exercise. To illustrate the underlying hand-object interactions, the *Bullet* hand avatar is also visualised (middle row). The sequence shows a user who first bumps into the target object (the green liquid dispenser) with the back of the fingers, which can be felt thanks to the whole-hand haptic rendering. After correctly positioning the hand, the virtual wall provides additional exercise-specific haptic feedback for the fingers. We found a transition period of $T_{tr} = 1\,\mathrm{s}$ to feel natural for the engagement of the virtual wall and $T_{tr} = 0.3\,\mathrm{s}$ for

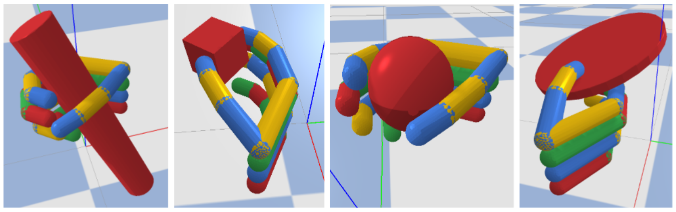

Fig. 5. Grasping of different free-floating objects. The objects were not constrained except for linear damping of 0.7 (in *Bullet*-specific units).

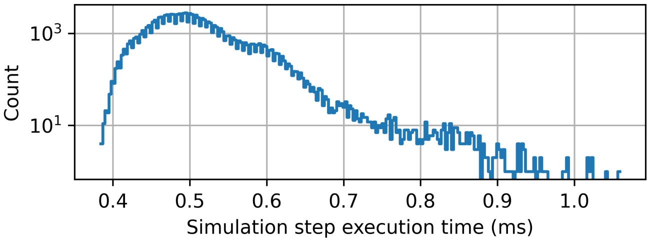

Fig. 6. Distribution of the loop execution time of 10^5 iterations during repeated collisions.

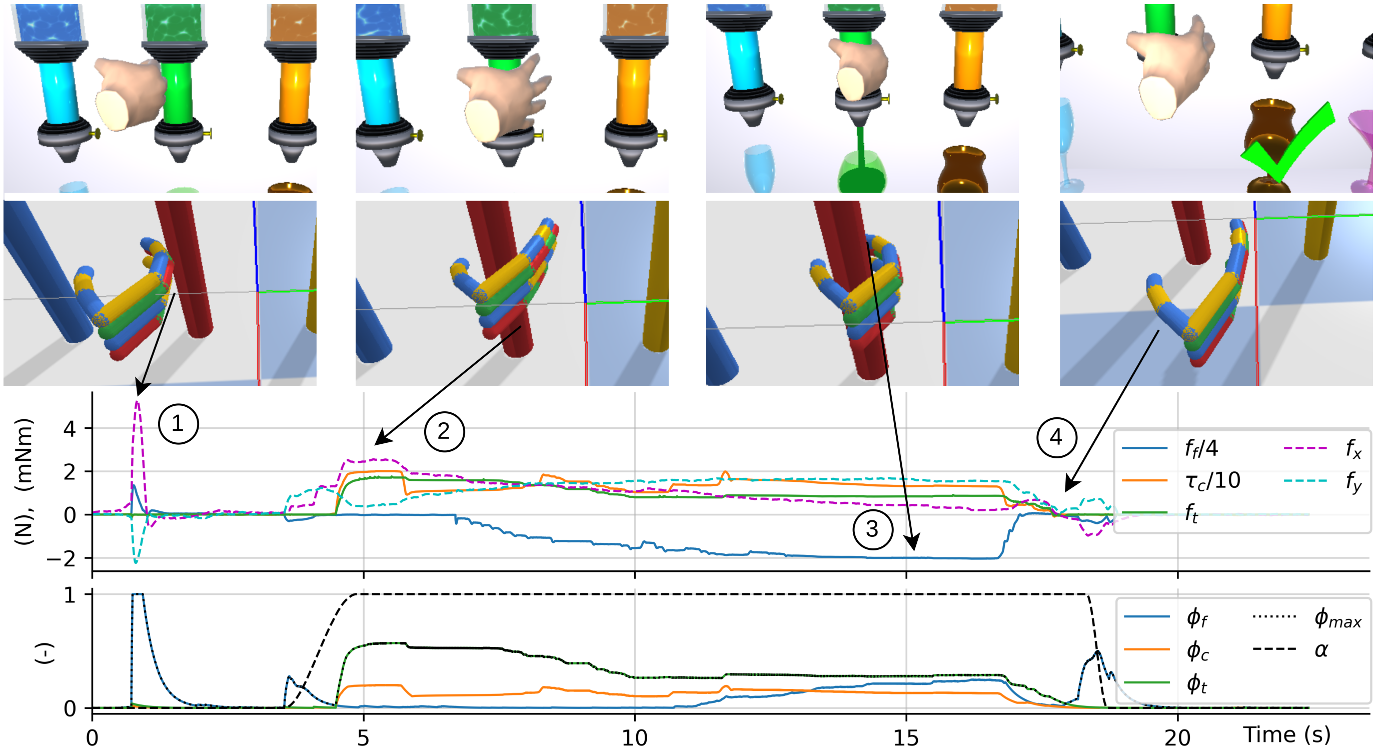

Fig. 7. Example of forces/torques and adaptive damping during physics-engine-based whole-hand rendering with addition of exercise-specific haptics. ①: The user first bumps into the liquid dispenser with the back of the fingers. Note the sharp increase in damping and the consecutive exponential decay thereof. ②: With a slight collision between hand and target, the user knows that the correct position is reached. Once the target has successfully been approached, also the exercise-specific haptics are faded in. ③:The user then carefully squeezes the liquid dispenser. ④: When the object is released upon task completion, the reverse transition to the physics engine-based rendering is triggered. Here, the user's fingers slightly collided with the object when retrieving the hand.

the disengagement. The addition of exercise-specific haptics to the physics engine-based whole-hand rendering achieves naturalistic and intuitive haptic grasping that can easily be tailored to the patient's needs during neurorehabilitation.

The advantage of our framework with respect to custom-tailored solutions (e.g., [13]–[15]) lies in the use of an off-the-shelf physics engine. This reduces the hurdle for implementation and at the same time increases flexibility as a physics engine is capable of simulating a wide variety of dynamic objects. In this context, we would like to mention that *Bullet* and other similar engines usually provide various parameters (e.g., error reduction parameter, maximum number of solver iterations, etc.) that could be tweaked for particular scenarios. However, in our approach, we endeavoured to modify only the minimal number of physics engine parameters possible (i.e., we only disable the inherent damping of objects) to ensure the transferability of the method to other physics engines.

As a next step, we will investigate whether and how our framework could be integrated as a package into game engines, e.g., in *Unity3D*. When larger numbers of objects or avatars have to be synchronized between the game engine and physics engine, a body-by-body synchronization might become cumbersome. Although there have been efforts for integrating haptic rendering in game engines (e.g., [34]–[36]), more research is needed in the special case of whole-hand haptic rendering.

IV. CONCLUSION AND FUTURE WORK

In this work, we demonstrated the use of an off-the-shelf physics engine for whole-hand haptic rendering in the context of a gamified neurorehabilitation exercise.

We foresee two directions for future research. On the technical side, stability criteria and the maximum admissible complexity of the rendered scene could be analyzed, and the integration with game engines should be facilitated. When it comes to neurorehabilitation research, the optimal complexity of the rehabilitation exercise (including task difficulty, number of objects, haptic sensations, etc) should be investigated from the perspective of neuroscience and clinical practice.

Finally, we are currently investigating the usability of the system, including the haptic device and the presented haptic rendering framework in a clinical setting in an ongoing study.

ACKNOWLEDGMENT

Many thanks go to Eduardo Villar Ortega, Karin Bütler and Alexandre Ratschat for their support during development. The constructive discussions and their willingness to repeatedly test the presented work are highly appreciated. We also thank Nathan Van Damme for his invaluable contribution to the therapeutic exercise development. Moreover, we thank Force Dimension for providing the hardware and their assistance in technical matters.

REFERENCES

[1] R. Bertani, C. Melegari, M. C. D. Cola, A. Bramanti, P. Bramanti, and R. S. Calabrò, "Effects of robot-assisted upper limb rehabilitation in stroke patients: a systematic review with meta-analysis," *Neurological Sciences*, vol. 38, pp. 1561–1569, 9 2017.

[2] R. Gassert and V. Dietz, "Rehabilitation robots for the treatment of sensorimotor deficits: a neurophysiological perspective," *Journal of NeuroEngineering and Rehabilitation*, vol. 15, p. 46, 12 2018.

[3] V. Klamroth-Marganska, J. Blanco, K. Campen, A. Curt, V. Dietz, T. Ettlin, M. Felder, B. Fellinghauer, M. Guidali, A. Kollmar, A. Luft, T. Nef, C. Schuster-Amft, W. Stahel, and R. Riener, "Three-dimensional, task-specific robot therapy of the arm after stroke: a multicentre, parallel-group randomised trial," *The Lancet Neurology*, vol. 13, pp. 159–166, 2 2014.

[4] N. Bolognini, C. Russo, and D. J. Edwards, "The sensory side of post-stroke motor rehabilitation," *Restorative Neurology and Neuroscience*, vol. 34, pp. 571–586, 8 2016.

[5] E. Basalp, P. Wolf, and L. Marchal-Crespo, "Haptic training: Which types facilitate (re)learning of which motor task and for whom answers by a review," *IEEE Transactions on Haptics*, vol. 1412, 2021.

[6] Özhan Özen, K. A. Buetler, and L. Marchal-Crespo, "Towards functional robotic training: motor learning of dynamic tasks is enhanced by haptic rendering but hampered by arm weight support," *Journal of NeuroEngineering and Rehabilitation*, vol. 19, p. 19, 12 2022.

[7] L. Wei, Z. Najdovski, H. Zhou, S. Deshpande, and S. Nahavandi, "Extending support to customised multi-point haptic devices in CHAI3D," *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, vol. 2014-Janua, no. January, pp. 1864–1867, 2014.

[8] F. Conti, F. Barbagli, R. Balaniuk, M. Halg, C. Lu, D. Morris, L. Sentis, J. Warren, O. Khatib, and K. Salisbury, "The chai libraries," in *Proceedings of Eurohaptics 2003*, Dublin, Ireland, 2003, pp. 496–500.

[9] I. A. Odermatt, K. A. Buetler, N. Wenk, Özhan Özen, J. Penalver-Andres, T. Nef, F. W. Mast, and L. Marchal-Crespo, "Congruency of information rather than body ownership enhances motor performance in highly embodied virtual reality," *Frontiers in Neuroscience*, vol. 15, 7 2021.

[10] C. S. Tzafestas, "Whole-hand kinesthetic feedback and haptic perception in dextrous virtual manipulation," *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans.*, vol. 33, pp. 100–113, 1 2003.

[11] N. Y. Lii, A. Pereira, J. Dietl, G. Stillfried, A. Schmidt, H. Beik-Mohammadi, T. Baker, A. Maier, B. Pleintinger, Z. Chen, A. Elawad, L. Mentzer, A. Pineault, P. Reisich, and A. Albu-Schäffer, "Exodex adam—a reconfigurable dexterous haptic user interface for the whole hand," *Frontiers in Robotics and AI*, vol. 8, 3 2022.

[12] G. Gonzalez-Badillo, H. I. Medellin-Castillo, T. Lim, J. M. Ritchie, R. C. Sung, and S. Garbaya, "A new methodology to evaluate the performance of physics simulation engines in haptic virtual assembly," *Assembly Automation*, vol. 34, pp. 128–140, 2014.

[13] C. Garre and M. A. Otaduy, "Toward Haptic Rendering of Full-Hand Touch," in *CEIG 09 - Congreso Espanol de Informatica Grafica*, C. Andujar and J. Lluch, Eds. The Eurographics Association, 2009.

[14] Q. Tong, Q. Wang, Y. Zhang, X. Liao, W. Wei, Y. Zhang, J. Xiao, and D. Wang, "Configuration-based Optimization for Virtual Hand Haptic Simulation," *IEEE Transactions on Haptics*, vol. 15, no. 3, pp. 613–625, 2022.

[15] Y. An, H. Cho, and J. Park, "Virtual whole-hand grasping feedback for object manipulation with a two-finger haptic interface," *Journal of Computing Science and Engineering*, vol. 14, pp. 41–51, 6 2020.

[16] D. Lobo, M. Sarac, M. Verschoor, M. Solazzi, A. Frisoli, and M. A. Otaduy, "Proxy-based haptic rendering for underactuated haptic devices," in *2017 IEEE World Haptics Conference (WHC)*, no. December 2018. IEEE, jun 2017, pp. 48–53.

[17] M. Verschoor, D. Lobo, and M. A. Otaduy, "Soft Hand Simulation for Smooth and Robust Natural Interaction," *25th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2018 - Proceedings*, pp. 183–190, 2018.

[18] A. Maciel, T. Halic, Z. Lu, L. P. Nedel, and S. De, "Using the physx engine for physics-based virtual surgery with force feedback," *International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 5, pp. 341–353, 2009.

[19] E. Ricardez, J. Noguez, L. Neri, D. Escobar-Castillejos, and L. Munoz, "Suturehap: Use of a physics engine to enable force feedback generation on deformable surfaces simulations," *International Journal of Advanced Robotic Systems*, vol. 15, 1 2018.

[20] C. E. Mower, T. Stouraitis, J. Moura, C. Rauch, L. Yan, N. Z. Behabadi, M. Gienger, T. Vercauteren, C. Bergeles, and S. Vijayakumar, "Ros-pybullet interface: A framework for reliable contact simulation and human-robot interaction," *arXiv*, 10 2022.

[21] M. Körber, J. Lange, S. Rediske, S. Steinmann, and R. Glück, "Comparing popular simulation environments in the scope of robotics and reinforcement learning," *arXiv*, 3 2021.

[22] J. Collins, S. Chand, A. Vanderkop, and D. Howard, "A review of physics simulators for robotic applications," *IEEE Access*, vol. 9, pp. 51 416–51 431, 2021.

[23] N. Van Damme, R. Rätz, and L. Marchal-Crespo, "Towards unsupervised rehabilitation: Development of a portable compliant device for sensorimotor hand rehabilitation," in *2022 International Conference on Rehabilitation Robotics (ICORR)*, 2022, pp. 1–6.

[24] Force Dimension. Lambda.7. [Online]. Available: https://www.forcedimension.com/products/lambda

[25] R. Rätz, F. Conti, R. M. Müri, and L. Marchal-Crespo, "A novel clinical-driven design for robotic hand rehabilitation: Combining sensory training, effortless setup, and large range of motion in a palmar device," *Frontiers in Neurorobotics*, vol. 15, pp. 1–22, 12 2021.

[26] R. Rätz, R. M. Müri, and L. Marchal-Crespo, "Design of a haptic palmar device with thumb flexion and circumduction movements for sensorimotor stroke rehabilitation," in *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2022, pp. 2644–2647.

[27] D. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE Transactions on Robotics and Automation*, vol. 9, pp. 624–637, 1993.

[28] R. Middleton and G. Goodwin, "Adaptive computed torque control for rigid link manipulations," *Systems & Control Letters*, vol. 10, pp. 9–16, 1 1988.

[29] J. Niiranen, "Fast and accurate symmetric euler algorithm for electromechanical simulations," 1999.

[30] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations*, 2nd ed. Berlin: Springer, 2006, iD: unige:12343.

[31] M. Kim, Y. Lee, Y. Lee, and D. Lee, "Haptic rendering and interactive simulation using passive midpoint integration," *The International Journal of Robotics Research*, vol. 36, pp. 1341–1362, 10 2017.

[32] SimBenchmark. Physics engine benchmark for robotics applications: RaiSim vs. Bullet vs. ODE vs. MuJoCo vs. DartSim. [Online]. Available: https://leggedrobotics.github.io/SimBenchmark

[33] P. Arcara and C. Melchiorri, "Control schemes for teleoperation with time delay: A comparative study," *Robotics and Autonomous Systems*, vol. 38, pp. 49–64, 2002.

[34] N. Balzarotti and G. Baud-Bovy, "Hpge: An haptic plugin for game engines," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11385 LNCS, pp. 330–339, 2019.

[35] D. Escobar-Castillejos, J. Noguez, R. A. Cárdenas-Ovando, L. Neri, A. Gonzalez-Nucamendi, and V. Robledo-Rella, "Using game engines for visuo-haptic learning simulations," *Applied Sciences (Switzerland)*, vol. 10, 7 2020.

[36] J. Rosskamp, H. Meißenhelter, R. Weller, M. O. Rüdel, J. Ganser, and G. Zachmann, "UnrealHaptics: Plugins for Advanced VR Interactions in Modern Game Engines," *Frontiers in Virtual Reality*, vol. 2, no. April, pp. 1–14, 2021.