DLR-IB-RM-OP-2020-85

Enhancement of a control approach to display high stiffness in virtual environments

Bachelorarbeit

Michael Rothammer

DLR Deutsches Zentrum
für Luft- und Raumfahrt

Institut für Robotik und Mechatronik

BJ.: 2020

IB.Nr.: DLR-IB-RM-OP-2020-85

BACHELORARBEIT

ENHANCEMENT OF A CONTROL APPROACH TO DISPLAY HIGH STIFFNESS IN VIRTUAL ENVIRONMENTS

Freigabe: Der Bearbeiter: Unterschriften

Michael Rothammer M. Rothammer

Betreuer:

Harsimran Singh

Thomas Hulin

Der Institutsdirektor

Prof. Alin Albu-Schäffer

Dieser Bericht enthält 83 Seiten, 50 Abbildungen und 0 Tabellen

Technical University of Munich

Michael Rothammer (03696542)

Enhancement of a control approach to display high stiffness in virtual environments

Thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science at the Department of Mechanical Engineering, Technical University of Munich

Supervisor(s):	PD Dr.-Ing. habil. Paul Kotyczka Chair of Automatic Control Prof. Dr.-Ing. Alin Albu-Schäffer German Aerospace Center, Institute of Robotics and Mechatronics
Advisor(s):	Dr. Harsimran Singh Dr.-Ing. Thomas Hulin
Submission date:	13 March 2020

Erklärung

Ich versichere hiermit, dass ich die von mir eingereichte Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Garching bei München, den 13. März 2020 ____________________ (Michael Rothammer)

Freiwillige ergänzende Erklärungen

☐ Mit der zeitlich unbefristeten Aufbewahrung meiner Arbeit in der Lehrstuhlbibliothek erkläre ich mich einverstanden.

☐ Ich stelle die Software, die ich im Rahmen dieser Arbeit entwickelt habe, dem Lehrstuhl für Regelungstechnik unter den Bedingungen der 3-Klausel-BSD-Lizenz zur Verfügung. Ich behalte dabei die sämtlichen Urheber- sowie Nutzungsrechte und werde als Autor namentlich genannt.

Garching bei München, den 13. März 2020 ____________________ (Michael Rothammer)

Chair of Automatic Control (Prof. Dr.-Ing. habil. Boris Lohmann)
Technical University of Munich
Boltzmannstraße 15
85748 Garching bei München
Germany

Lehrstuhl für Regelungstechnik (Prof. Dr.-Ing. habil. Boris Lohmann)
Technische Universität München
Boltzmannstraße 15
85748 Garching bei München
Deutschland

Task Description

The Institute of Robotics and Mechatronics researches new robotic technologies for applications in space and on Earth. Your mission is to extend an existing control approach for rendering high stiffness in haptics from one to multiple degrees of freedom (DoF). The resulting controller should be able to realistically display very stiff virtual objects without distorting the interaction through instability.

Your main tasks:
- Extend a 1-DoF haptic control approach to multiple degrees of freedom
- Implement and test the controller on a suitable haptic device
- Compare the controller to existing state-of-the-art control approaches
- Optional: Evaluate the controller in a user study

Abstract

Haptic feedback is a powerful tool to display virtual environments in a detailed and immersive way. However, stably realizing virtual objects with high stiffnesses remains a difficult task, mainly due to time discretization.

The Successive Force Augmentation approach is based on a concept of progressively changing the displayed force in order to achieve a desired stiffness. Primarily based on a low stiffness, the force is successively adapted during the interaction until the desired stiffness is reached.

This thesis covers the enhancement of the Successive Force Augmentation approach using alternative methods to update the displayed force to reduce response time, a passivity controller to ensure stability as well as a filter to avoid discontinuities in the displayed force. Furthermore, the enhanced approach is then extended to three translational dimensions. For validation, multiple experiments are conducted to compare different implementations with each other and the relevant haptic control approaches Time-domain Passivity Approach and Force Bounding Approach.

Kurzfassung

Haptisches Feedback ist ein mächtiges Mittel zur detailgetreuen und immersiven Darstellung von virtuellen Umgebungen. Die stabile Realisierung von virtuellen Objekten mit hohen Steifigkeiten ist allerdings, vor allem bedingt durch Zeitdiskretisierung, sehr schwierig.

Der Ansatz der Successive Force Augmentation basiert auf dem schrittweisen Anpassen der dargestellten Kraft, um auf stabile Weise eine gewünschte Steifigkeit darzustellen. Dazu wird zunächst von einer geringen Steifigkeit ausgegangen. Während der Interaktion wird dann sukzessive die dargestellte Kraft verändert, bis die gewünschte Steifigkeit dargestellt wird.

Diese Arbeit behandelt die Erweiterung der Successive Force Augmentation mit alternativen Methoden zur Anpassung der dargestellten Kraft für eine kürzere Antwortzeit, einem Passivitätsregler zur Stabilitätssicherung sowie einem Filter zur Vermeidung von unstetigen Kräften. Außerdem wird der Ansatz von einem auf drei translationale Freiheitsgrade erweitert. Um die Ergebnisse zu validieren, werden in zahlreichen Experimenten verschiedene Implementierungen miteinander und mit den maßgeblichen haptischen Regelungsmethoden Time-domain Passivity Approach und Force Bounding Approach verglichen.

Acknowledgments

I would like to express my great appreciation to Dr. Harsimran Singh and Dr.-Ing. Thomas Hulin for their invaluable support throughout my work on this thesis.

Furthermore, I am very grateful to PD Dr.-Ing. habil. Paul Kotyczka and Prof. Dr.-Ing. Alin Albu-Schäffer for their supervision and the opportunity to pursue this thesis at the Institute of Robotics and Mechatronics of the German Aerospace Center.

Contents

Chapter 1

Introduction

1.1 Haptics

Electronic devices are powerful tools created for widely varying purposes. One of the most important usages for electronics is to provide the user with information. To achieve that, they need to address at least one of the human senses.

Hearing is a very important one of these senses, especially considering the purpose of conveying information. Humans typically communicate through speech, so in an acoustic way. Therefore, the invention of the loudspeaker around 1861 by Johann Philipp Reis according to Gifford [9] marks an important milestone for human-machine interaction.

According to Martin [17], one of the first electronic devices to provide information visually was a cathode ray tube in 1897. This was another enormous achievement and especially useful because human perception relies heavily on vision.

The sense of touch is another very important channel for humans to experience their environment. Haptic technology makes use of this by conveying information to a human through physical stimulation. Haptics are generally divided into tactile and kinaesthetic feedback.

1.1.1 Tactile feedback

The idea of tactile feedback is to stimulate the user's skin to provide information using primarily the cutaneous sensory system. This can be done in a variety of different ways and for a variety of different purposes. A very widespread application of tactile feedback are basic vibrations used in everyday technology such as cellphones or video game controllers. Even though those signals are comparatively easy to generate, they can considerably enhance the interaction by using an additional human sense.

Vibrational feedback has been examined for many different applications, for example in a vest for dogs for an additional channel of communication (Golan et al. [10]). A very impressive application of vibrotactile feedback include the display of textures through vibrations of controlled amplitude and frequency as described by Lin and Smith [16]. A computer mouse with the ability to display textures has been developed by Kyung et al. [13]. Park and Kim [20] achieved a flexible vibrotactile actuator with the intent of implementing it in wearable devices.

Another impressive work on this topic has been done by Yoshino and Shinoda [29], who realized contactless tactile feedback.

To make use of the spatial resolution of the human perception on the fingertip, multiple contact elements arranged in an array have been used. Different methods of actuation for these elements have been evaluated ranging from electromagnets to shape-memory alloys (Yang et al. [26], Yang et al. [27], Yang et al. [28], Velázquez et al. [25], Moy et al. [19]). Those pin arrays can provide different stimuli, such as braille, by activating the elements systematically. However, they generally come with the drawback of not being wearable.

A lot of work covers the application of forces on the human fingerpad. Damian et al. [5] have developed a wearable device which can display both tangential and normal forces on the skin even though it uses just a single actuator. Kim et al. [12] achieved contact force feedback in three degrees of freedom (DoF) and finger rotation tracking in a single wearable device. In addition to tangential and normal force, Lim et al. [15] implemented a 3-DoF tip-tilt mechanism to enable 5 DoF in total on a device that is still wearable. Singh et al. [22] have developed a ferro-fluid based device that can display contact orientation and texture simultaneously, thereby combining vibrational and force-based tactile feedback.

1.1.2 Kinaesthetic feedback

Haptic feedback that primarily uses the human proprioception to convey information is known as kinaesthetic feedback. A typical way of implementing kinaesthetic feedback is by exerting forces on the user. Usually, this is done with a haptic interface, the device generating those forces. These interfaces are technically comparatively complex, which is why kinaesthetic feedback is not yet as widely spread as tactile feedback.

However, it is a very important tool to utilize additional channels to deliver information to a user. In some applications, kinaesthetic feedback can make interactions considerably more intuitive. A typical example of this is telerobotics, which involves remotely controlled robots. Additionally to visual feedback in the form of live camera feeds, the operator can be provided information about the controlled robot and its interaction with the environment through kinaesthetic feedback. For instance, the forces acting on the end effector of the controlled robot can be displayed to the operator and vice versa, so the interaction is very intuitive. An impressive example of how effective this can be is Analog-1, which lacks academic publishing so far, but has been described by ESA and DLR [6], a project of the european space agency and the german aerospace center that concluded with an experiment that consisted of remote rock sampling on earth from the ISS, supported by kinaesthetic feedback.

Another impressive application of telerobotics is telesurgery. As the name suggests, this field addresses remotely operated surgical procedures using telerobotics. The main benefit of this technology is that expert surgeons can operate on a patient in need of their knowledge and experience without the need of extensive travel. Here, kinaesthetic feedback can vastly improve the surgeon's perception of the patient and the ongoing processes. For instance, the patient's tissue can be palpated by the surgeon providing additional information about its properties as described by Mayer et al. [18].

Kinaesthetic feedback can also be applied in the training of prospective surgeons, as covered by Escobar-Castillejos et al. [7], and other professionals dealing with similarly sensible tasks such as dentistry (Bogoni et al. [3], Zhang et al. [30]), but also simpler ones like grinding (Balijepalli and Kesavadas [2]), and even lock picking (Arthur et al. [1]) with virtual reality. They can get hands-on education on very important and responsible missions without actually facing serious consequences in the case of mistakes. Apart from training experts, kinaesthetic feedback can generally be a valuable addition to virtual reality as it considerably improves immersion.

This thesis focuses on the kinaesthetic form of haptic feedback and how to solve some of its most essential challenges.

1.2 Open research challenges in haptics

By providing haptic feedback to a user, a closed feedback loop (illustrated in Fig. 1.1) is created, because the force exerted on the user will affect the position of the haptic interaction point (HIP) which in turn is used to compute the forces to display. The HIP is the point where a user interacts with the haptic interface, for example the end effector of a robot.

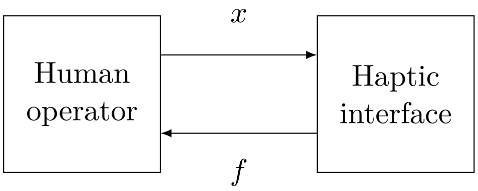


Figure 1.1: The position x of the HIP is affected by the human operator. Depending on this position, the haptic interface displays a resulting kinaesthetic feedback force f. This force proceeds to affect the human operator and the HIP position as a consequence.

1.2.1 Quantization error

When the position of the HIP is measured in order to compute haptic feedback forces accordingly, the exact value can almost certainly not be perfectly represented digitally. As Fig. 1.2a shows, there are only certain, usually equidistant values that the measurement can result in, each assigned a specific interval of analogous signal.

Also, there might be inaccuracy of the originally collected data itself due to limitations in the sensor's resolution. Therefore, the measured position (x_{me}) is very likely to be different than the actual position (x):

$\exists \quad x \quad \text{satisfying}$

$$x_{\mathrm{me}}(x) \neq x \tag{1.1}$$

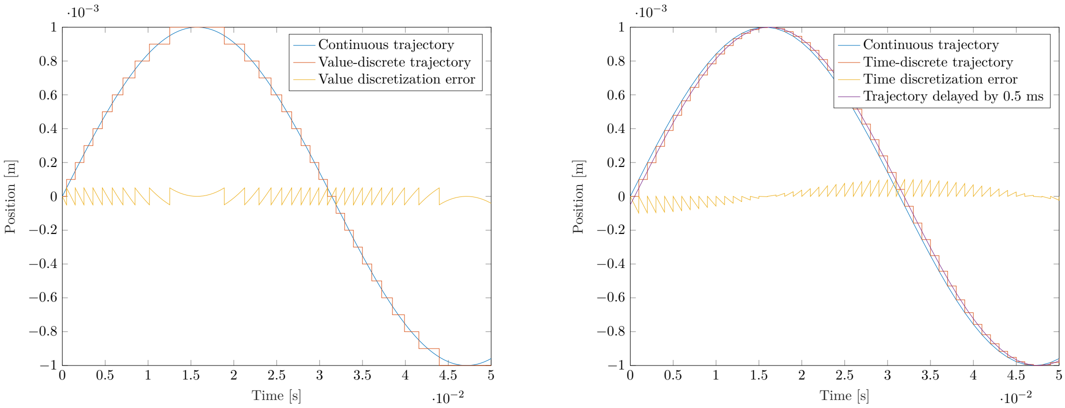
1.2.2 Sampling error

When the HIP penetrates a virtual object, this will only be measured once the next sample is taken. The maximum possible time delay between the HIP entering the virtual object (or moving in general) and this process being measured (T_d) can be computed using the sample rate (f_s):

$$0 \leq T_\mathrm{d} < \Delta T = \frac{1}{f_\mathrm{s}} \tag{1.2}$$

On average, the measured values are delayed by $\frac{1}{2}\Delta T$. Fig. 1.2b shows that the measured signal follows the original trajectory, but thereby is closer to the same signal delayed by $\frac{1}{2}\Delta T$.

After the position has been measured, the resulting haptic feedback force needs to be computed, which itself adds another time delay. For simple virtual environments however, this delay is negligible compared to that one generated by sampling.



(a) The quantized (resolution: $\Delta x = 0.1mm$) position and its difference to the actual value.

(b) The time-discrete (sampling frequency: $f_\mathrm{s} = 1000Hz$) position and its difference to the actual value.

Figure 1.2: The different forms of discretization and their respective resulting inaccuracies.

Fig. 1.2b shows clearly how the gradient of the signal directly affects the sampling error's effect. With a steeper gradient, the signal will change more in a given amount of time. Because this change directly represents the sampling error until the next sample is taken, the sampling error itself can grow further in the sampling time. Not only the absolute value, but also the sign of the sampling error result from its gradient. Whenever the input signal is falling, the measured values will always be greater than or equal to the actual signal, so the error will be positive, whereas a rising continuous signal will lead to a negative error.

1.2.3 Inaccuracy

In the real implementation, both sampling errors and quantization errors appear.

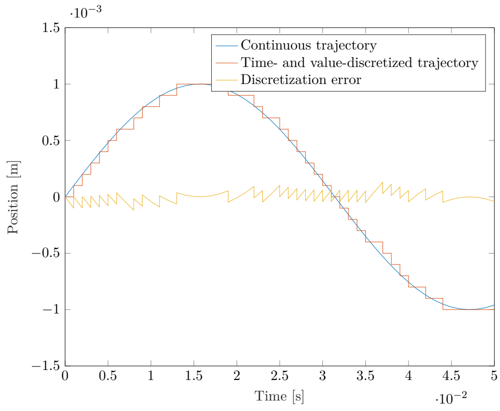


Figure 1.3: The observed position and its difference to the actual value.

As a result, the haptic feedback force is computed based on a position that is probably different from the actual position of the HIP. For example, if the HIP is moving into a virtual object, the user might feel the feedback force only after some delay. At this time, the HIP might have already moved even further into the virtual object, but before the user feels the accordingly increased force, some more time will have passed. Overall, the user will feel smaller forces than expected when penetrating the object.

Only once the HIP stops moving, the forces will "catch up". As soon as it starts moving back out of the virtual object, the measured position will again lag behind, in this case staying further inside the virtual object than the HIP actually is. As a result, the displayed forces will be higher than their ideal values.

This effect depends highly on the sampling frequency, with a higher frequency resulting in smaller differences between the displayed and the "correct" haptic feedback force. The quantization error's amplitude is independent of the signal gradient, so it does not contribute to this observed delay.

1.2.4 Passivity condition

A very fruitful approach to ensuring stability is to limit the energy generated by a virtual environment. This energy can be easily computed using the conjugate flow and effort variables velocity and force.

By convention, energy introduced into the haptic device by the operator while a virtual object is being penetrated is considered positive. While the HIP's motion is directed back out of a virtual object, the operator absorbs energy. Therefore, this energy is considered negative.

This means that the energy can be regarded as potential energy that increases while the operator pushes into a virtual object, and decreases again when the HIP leaves it again. Assuming an initial value of $E(0) = 0$, an interaction is denoted as passive if the energy stays greater than zero for all subsequent samples:

$$E(n) > 0 \quad \forall \quad n \tag{1.3}$$

In order to visualize if the passivity condition is properly satisfied, the position-force diagram is very effective. It shows the force displayed by the haptic device with respect to the position of the HIP (naturally, this diagram usually only displays a single degree of freedom).

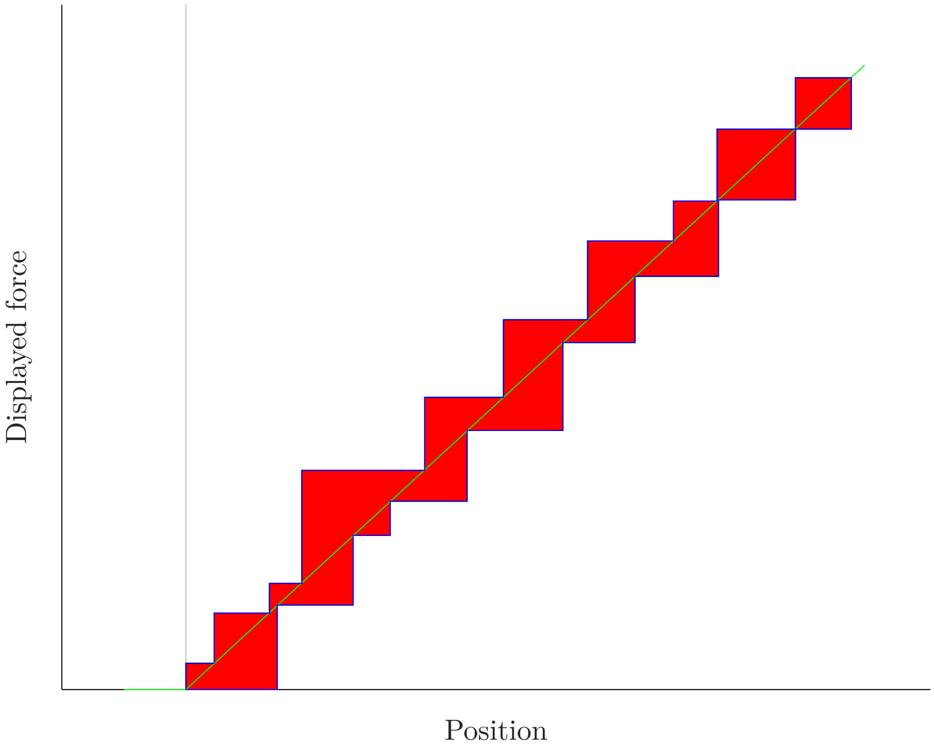


Figure 1.4: Schematic representation of a position-force diagram with randomly generated data. The red area can be interpreted as the energy generated by the virtual environment.

The energy E introduced and dissipated due to the haptic feedback forces can be very conveniently computed as the integral of the displayed Force with respect to the position of the HIP as shown in (1.4a). This integral can be visualized as the area between the $F = 0$ -axis and the graphed values for position and force. Because the signals used in this computation are discrete, the integral translates to a sum, which can also be calculated as a recursive sequence:

$$E(t) = \int_{x(t_0)}^{x(t)} F(t)\delta x \tag{1.4a}$$

$$E(n) = \sum_{i=1}^{n} F(i) \cdot \dot{x}(i) \cdot \Delta T \tag{1.4b}$$

$$E(0) = 0 \tag{1.4c}$$

$$E(n) = E(n-1) + F(n) \cdot \dot{x}(n) \cdot \Delta T \tag{1.4d}$$

$$\tag{1.4e}$$

Ideally, the positive work done during the motion directed into an virtual object and the negative work done moving back outside of it would cancel out, so the total energy would still be zero. However, discretization and measurement errors might lead to differences in these values, so the total energy might not be zero.

Looking at the graphical representation of the energy as an area as displayed in Fig. 1.4, the total energy generated or dissipated by an interaction is represented by the area in between the trajectory of the inbound motion and that one of the movement back outside of the virtual object.

For passivity, the delay phenomenon described in § 1.2.3 plays a central role. Assuming a virtual object that displays opposing forces following a strictly monotonically increasing function depending on the penetration depth, the displayed force would increase with increasing penetration depth. Because the measured position is delayed compared to the actual value, the displayed forces are generally smaller than their ideal values while the HIP moves into the virtual object. Subsequently, the energy absorbed by the haptic device as potential energy is smaller than it would ideally be.

For the outbound motion, the measured position is again delayed. In this case, the displayed forces are belatedly reduced, so they are mostly too high. This results in the energy output by the haptic device being greater than the ideal value.

Overall, the absorbed energy is smaller, whereas the energy output by the device is greater than in the ideal case as a consequence of sampling errors. This can be seen clearly in Fig. 1.4 and is a perfect example of a violation of the passivity condition and it represents the most fundamental challenge of kinaesthetic feedback.

1.2.5 Rate-hardness

A relevant metric for haptics is the so-called rate-hardness, which is defined by Lawrence et al. [14] as the ratio of the rate of change of the displayed force ($\dot{F}$) to the velocity ($\dot{x}$) of the HIP for the initial contact.

$$H_{\mathrm{R}} = \frac{\dot{F}}{\dot{x}} \tag{1.5}$$

Interestingly, the unit for this metric is $\frac{\frac{N}{s}}{\frac{m}{s}}$, so it could be easily simplified to $\frac{N}{s}$, which is generally used to measure the stiffness of springs. However, Lawrence et al. [14] have shown that rate-hardness is closer to the human perception than traditional stiffness.

1.2.6 Virtual spring

The original and most straightforward approach to display a virtual object is to simply determine if the user has penetrated the object, and if so, compute an outbound force F_{h} according to a virtual spring with a stiffness k_{v} attached to the surface x_{s} of the object and the user as illustrated by Fig. 1.5. Unfortunately, this approach is not very robust against sampling and quantization errors.

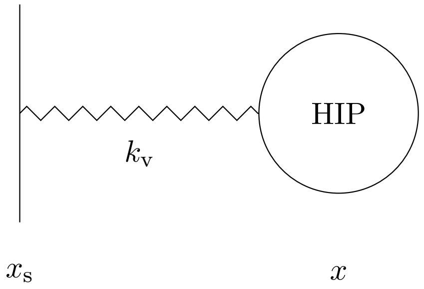


Figure 1.5: Virtual spring

$$F(n) = k_\mathrm{v} \cdot (x(n) - x_\mathrm{s}(n)) \tag{1.6}$$

This lack of robustness could be addressed by introducing significant virtual damping in the model. However, this would distort the human perception of the object, so it is preferably avoided. In particular, it was found to increase the perceived hardness of a virtual object while the user is in direct contact with it, but to decrease it when the user makes contact with it (van Beek et al. [24]). Therefore, it reduces the accuracy of the conveyed hardness and thereby decreases transparency.

This approach's sensitivity to measurement errors (x_e) is related to the linear dependency between the position and the resulting haptic feedback force:

$$F(n) = k_\mathrm{v} \cdot (x(n) + x_\mathrm{e}(t) - x_\mathrm{s}(n)) \tag{1.7a}$$
$$F_\mathrm{e}(t) = k_\mathrm{v} \cdot x_\mathrm{e}(t) \tag{1.7b}$$

The equations show that the virtual spring's stiffness directly scales the error in the haptic feedback force (F_e). If this stiffness is set to a very high value, this can easily lead to improper forces and eventually to instability.

The maximum stiffness at which discretization effects do not lead to instability , k_c, can be calculated according to Colgate et al. [4] depending on the inherent physical damping (c_i) of the haptic interface caused by mechanical inefficiencies as well as the sampling time (ΔT):

$$k_\mathrm{c} = \frac{2 \cdot c_i}{\Delta T} \tag{1.8}$$

Chapter 2

Introduction to selected haptic control approaches

2.1 Time-domain Passivity Approach

The time-domain passivity approach (TDPA) relies heavily on the concept of the passivity condition. In particular, the energy stored in the system is monitored constantly. This is done using a passivity observer that computes the current energy according to (1.4a).

When it comes to determining the haptic feedback forces, the energy generated (or dissipated) as a result of this force is examined first. After computing the force F_{VE} according to an arbitrary virtual environment, the system's preliminary energy, E_{pr}, is calculated:

$$E_{\mathrm{pr}}(n) = E(n-1) + F_{\mathrm{VE}}(n) \cdot \dot{x}(n) \cdot \Delta T \tag{2.1}$$

In order to keep the system stable, it is then verified that it is still passive after applying the force computed by the virtual environment. If the preliminary value for the energy is still positive, the haptic feedback force can be stably displayed the way it has been computed, whereas a negative value suggests exerting this force would lead to activity. In the latter case, an additional force F_{d} to compensate (i. e. dampen) that excess energy generated by the haptic device is calculated:

$$F_{\mathrm{d}}(n) = \begin{cases} 0 & \text{if } E_{\mathrm{pr}}(n) \geq 0 \\ -\frac{E_{\mathrm{pr}}(n)}{\dot{x}(n) \cdot \Delta T} & \text{if } E_{\mathrm{pr}}(n) < 0 \end{cases} \tag{2.2}$$

The force F_{TDPA} to actually be displayed at the HIP is then calculated by simply adding up the force from the virtual environment and the compensating force:

$$F_{\mathrm{TDPA}}(n) = F_{\mathrm{VE}}(n) + F_{\mathrm{d}}(n) \tag{2.3}$$

After F_{TDPA} has been calculated, the final value for the observed energy can be determined:

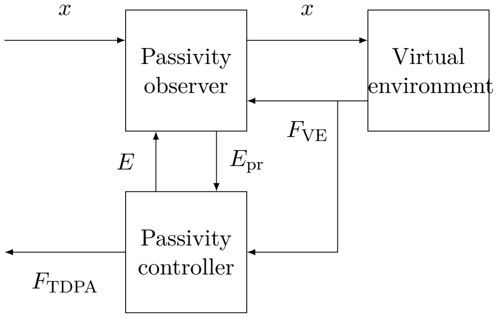
$$E(n) = E(n-1) + F_{\mathrm{TDPA}}(n) \cdot \dot{x}(n) \cdot \Delta T \tag{2.4}$$

Note that this equation yields the exact same result as (2.1) if the compensating force does not need to be applied.

The compensating force can be mathematically shown to keep the system stable:

$$\begin{aligned} E(n) &= E(n-1) + F_{\mathrm{VE}} \cdot \dot{x}(n) \cdot \Delta T + F_{\mathrm{d}} \cdot \dot{x}(n) \cdot \Delta T \\ &= E_{\mathrm{pr}}(n) + F_{\mathrm{d}} \cdot \dot{x}(n) \cdot \Delta T \\ &= \begin{cases} E_{\mathrm{pr}}(n) & \text{if } E_{\mathrm{pr}}(n) \geq 0 \\ E_{\mathrm{pr}}(n) - \frac{E_{\mathrm{pr}}(n)}{\dot{x}(n) \cdot \Delta T} \cdot \dot{x}(n) \cdot \Delta T = 0 & \text{if } E_{\mathrm{pr}}(n) < 0 \end{cases} \end{aligned} \tag{2.5}$$

The entire TDPA can be summarized in a block diagram showing the information flow:



[H]

Figure 2.1: The position x is fed to the virtual environment to obtain F_{VE}. Then, the passivity observer calculates E_{pr} from that. This value is used by the passivity controller to compute F_{d} if applicable. From this, F_{TDPA} can be determined, which is the resulting force, and also used to find E. The passivity observer will use this value when calculating E_{pr} in the following sample.

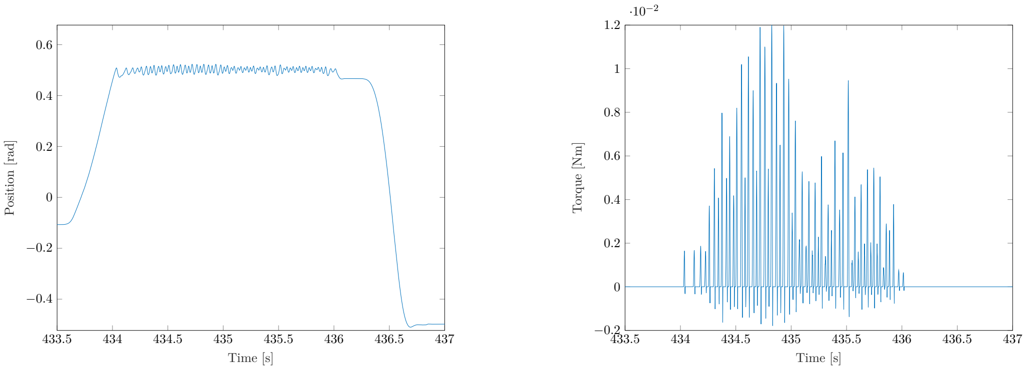
In real systems, the energy observation could respect the inherent physical damping of the system (c_{i}). For this, an additional term is added to its equation:

$$E_{\mathrm{pr}}(n) = E(n-1) + F_{\mathrm{VE}}(n) \cdot \dot{x}(n) \cdot \Delta T + \dot{x}^2(n) \cdot c_{\mathrm{i}} \cdot \Delta T \tag{2.6a}$$

$$E(n) = E(n-1) + F_{\mathrm{TDPA}} \cdot \dot{x}(n) \cdot \Delta T + \dot{x}^2(n) \cdot c_{\mathrm{i}} \cdot \Delta T \tag{2.6b}$$

For practical implementation, it is useful to reset the energy observation once the HIP has left the virtual object. If this is not done, positive energy will accumulate outside of the object due to the consideration of internal damping, which can then allow temporary instabilities once the user enters the virtual object afterwards.

Generally, implementations of the TDPA do not consider the internal damping. This ensures the passivity even more, as an incorrectly determined or a changing value for c_{i} could potentially allow activity as a result of erroneously computed energy.

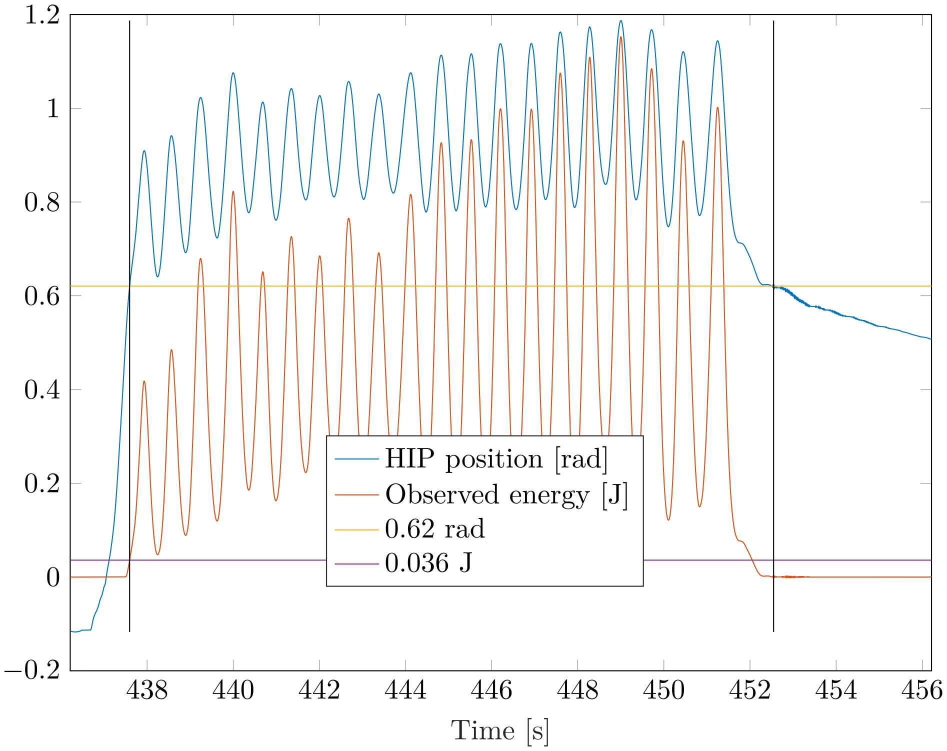


(a) The TDPA can produce perceptible oscillations.

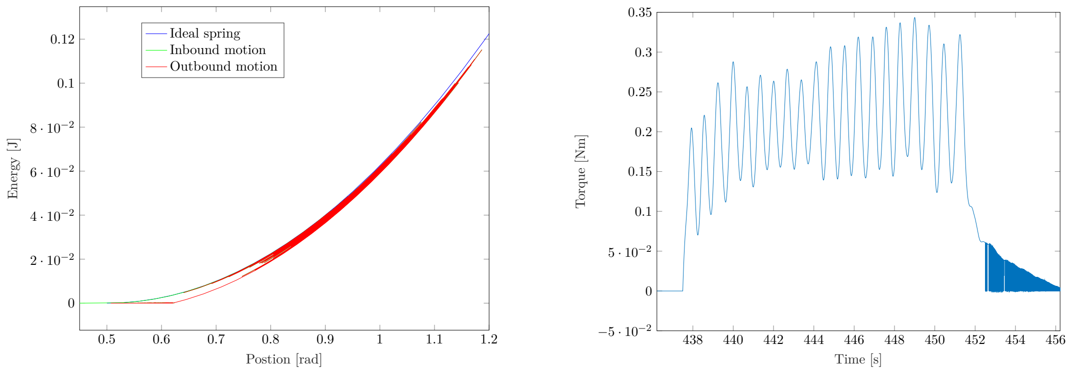
(b) The adaptive damping is responsible for sudden jumps in force to dissipate excess energy.

Figure 2.2: Oscillation phenomenon with TDPA after the stored energy reaches zero inside a virtual object

Unfortunately, TDPA comes with the drawback of showing some oscillation phenomena as shown in Fig. 2.2. This happens not only on contact, but also after leaving an object after interacting with it as described in Fig. 2.3. In that case, the energy stored in the virtual spring from the inbound motion would quickly be fed back into the system due to discretization errors as explained in § 1.2.4. However, as soon as the observed energy reaches negative values, the adaptive damping engages and significantly alters the displayed force. Because this damping is calculated to exactly dissipate all of the excess energy, the value will be zero or close to it in the next sample, setting the force back to an undamped value. At this point, the passivity controller is very susceptible to further energy loss and the adaptive damping will engage accordingly easily. This closes the loop that will be perceived as unpleasant oscillations of the HIP. This mechanism is of course more pronounced if the stiffness is high because of the higher quantization-based energy losses, but it will engage regardless of the stiffness as soon as the observed energy reaches zero.



(a) The HIP position and the resulting observed energy over time. For the same position, the energy observer returns a lower value after some energy has been lost to discretization during the interaction. In fact, the value of the observed energy reaches zero while the HIP is still inside the virtual object by 0.12 rad. Note that the energy values are not to scale, but increased by a factor of 10. The low-frequency oscillations of the HIP position are intentional movement by the operator.



(b) The observed energy over the HIP's position. It is clear how the value is initially very close to the ideal case, but as more energy gets lost to quantization, the value successively decreases. This finally results in the zero point effectively moving into the virtual object.

(c) As a result of the missing energy that has been lost to discretization, the displayed torque is heavily affected by the adaptive damping starting at around 452.2*s*.

Figure 2.3: Oscillations form TDPA after the computed energy reaches zero during the outbound motion.

2.2 Force Bounding Approach

The force bounding approach (FBA) as described by Kim et al. [11] is based on the energy approach as well. Hence, the energy is observed in a similar way as in the TDPA, according to (2.6a). Note that the internal damping c_i is respected in FBA. Again, the haptic feedback force is first calculated according to an arbitrary virtual environment. Then, the resulting force is constrained to not exceed a certain magnitude:

$$|F_\mathrm{max}|(n) = \sqrt{\frac{4 \cdot E(n) \cdot c_\mathrm{i}}{\Delta T}} \tag{2.7a}$$

$$F_\mathrm{constrained}(n) = \begin{cases} |F_\mathrm{max}|(n) & \text{if } F_\mathrm{VE}(n) > |F_\mathrm{max}|(n) \\ F_\mathrm{VE}(n) & \text{if } -|F_\mathrm{max}|(n) < F_\mathrm{VE}(n) < |F_\mathrm{max}|(n) \\ -|F_\mathrm{max}|(n) & \text{if } F_\mathrm{VE}(n) < -|F_\mathrm{max}|(n) \end{cases} \tag{2.7b}$$

This boundary is not chosen arbitrarily, but in order to satisfy the following passivity condition:

$$0 \leq E(n) = \dot{x}(n)^2 \cdot \Delta T \cdot c_\mathrm{i} + F_\mathrm{FBA}(n-1) \cdot \dot{x}(n) \cdot \Delta T + E(n-1) \tag{2.8}$$

This condition can be rewritten as follows:

$$\begin{aligned} E(n) &= E(n) - E(n-1) + \frac{1}{4} \cdot F_\mathrm{FBA}(n-1)^2 \cdot \frac{\Delta T}{c_\mathrm{i}} \\ &+ E(n-1) - \frac{1}{4} \cdot F_{FBA}(n-1)^2 \cdot \frac{\Delta T}{c_\mathrm{i}} \\ &= \dot{x}(n)^2 \cdot \Delta T \cdot c_\mathrm{i} + \frac{1}{4} \cdot F_\mathrm{FBA}(n-1)^2 \cdot \frac{\Delta T}{c_\mathrm{i}} + F_\mathrm{FBA}(n-1) \cdot \dot{x}(n) \cdot \Delta T \\ &+ E(n-1) - \frac{1}{4} \cdot F_\mathrm{FBA}(n-1)^2 \cdot \frac{\Delta T}{c_\mathrm{i}} \\ &= \frac{1}{4} \cdot (2 \cdot \sqrt{\Delta T \cdot c_\mathrm{i}} \cdot \dot{x}(n) + \sqrt{\frac{\Delta T}{c_\mathrm{i}}} \cdot F_\mathrm{FBA}(n-1))^2 \\ &+ E(n-1) - \frac{1}{4} \cdot F_\mathrm{FBA}(n-1)^2 \cdot \frac{\Delta T}{c_\mathrm{i}} \\ \text{Let } y &= 2 \cdot \sqrt{\Delta T \cdot c_\mathrm{i}} \cdot \dot{x}(n) + \sqrt{\frac{\Delta T}{c_\mathrm{i}}} \cdot F_\mathrm{FBA}(n-1) \\ E(n) &= \frac{1}{4} \cdot y^2 + E(n-1) - \frac{1}{4} \cdot F_\mathrm{FBA}(n-1)^2 \cdot \frac{\Delta T}{c_\mathrm{i}} \end{aligned} \tag{2.9}$$

The energy will always be positive ($E(n) \geq 0$), if:

$$E(n-1) - \frac{1}{4} \cdot F_\mathrm{FBA}(n-1)^2 \cdot \frac{\Delta T}{c_\mathrm{i}} \geq 0 \tag{2.10}$$

The reason for this is that $\frac{1}{4} \cdot y^2 \geq 0 \quad \forall \quad y \in \mathbb{R}$. Assuming positive and real internal damping and sampling time, y will always be real. Therefore, the maximum absolute value for the force can be computed as done in (2.7a).

Again, it is very useful to reset the accumulated energy once the HIP leaves the virtual object to avoid instabilities.

A very important factor deciding the effectiveness of the FBA is to set the correct value for c_i. If this value is chosen too high, the system's energy is computed incorrectly. In particular, the dissipation term ($\dot{x}^2 \cdot \Delta O \cdot c_\mathrm{i}$) in (2.6a) will return higher values. Additionally, according to (2.7a), the maximum and minimum allowed forces are absolutely greater than they would normally be which can in turn lead to discretization-induced activity.

On the other hand, a value that is too low will reduce the approach's effectiveness as well. In this case, the limits for the force are smaller than actually necessary to ensure stability, so the transparency is reduced.

Ultimately, it is obvious that c_i needs to be set precisely to produce the best results. Fortunately, by depending considerably on this parameter, FBA can be used to find the ideal value very nicely. The following algorithm explains the method in detail.

Algorithm 2.1: Find c_i in N steps with an accuracy of $2^{-N} \cdot (c_{\mathrm{i,\ high,\ initial}} - c_{\mathrm{i,\ low,\ initial}})$

```
Result: $c_i$
begin
    for $i = 1 : 1 : N$ do
        $c_{\mathrm{i,\ mean}} = \frac{c_{\mathrm{i,\ high}} + c_{\mathrm{i,\ low}}}{2}$
        test stability for $c_{internal,mean}$
        if $c_{i,\ mean}$ leads to stable interaction then
            $c_{\mathrm{i,\ low}} = c_{\mathrm{i,\ mean}}$
        else
            $c_{\mathrm{i,\ high}} = c_{\mathrm{i,\ mean}}$
    $c_\mathrm{i} = c_{\mathrm{i,\ high}}$
```

Experiments show that in any case, FBA will effectively constrain the displayed virtual stiffness to a value smaller than or equal to k_c according to (1.8). This is reasonable considering that this is the maximum stiffness that can be stably displayed without an additional controller, so the FBA limits the displayed stiffness to that value.

2.3 Successive Force Augmentation

The successive force augmentation (SFA), as explained by Singh et al. [21] and Singh et al. [23] avoids instability by using a low stiffness k_{v} that is stable even when assuming no virtual damping and without any additional controller.

In order to still display high stiffness ($k_{\mathrm{de}} > k_{\mathrm{v}}$), the deformation of the virtual spring is artificially increased by an offset $O(n)$ while the user is interacting with the virtual environment. The following equation illustrates how the displayed force f is calculated when a virtual object's surface, located at x_{s}, is penetrated:

$$f(n) = \begin{cases} -k_{\mathrm{v}} \cdot (e(n) + O(n)) & \text{if } e(n) > 0 \\ 0 & \text{else} \end{cases} \tag{2.11}$$

where

$$e(n) = x(n) - x_{\mathrm{s}}(n) \tag{2.12}$$

The actually displayed stiffness can be determined in order to evaluate the validness of the offset as shown in (2.13). However, especially for a high desired stiffness, this equation can get very close to a singularity when $e(n)$ is very small. With the previously presented offset, the HIP might leave the original virtual object while still being pulled further due to the additional spring deformation.

$$k_{\mathrm{di}} = -\frac{f(n)}{e(n)} \quad \forall e(n) > 0 \tag{2.13}$$

The additional deformation $O(n)$ is updated every time the HIP's motion's direction changes from out of the virtual object to into it according to (2.14a). The change in this offset is computed to display the critical stiffness k_{c} as derived from (1.8) at the beginning of the inwards motion, so the discretization effects can be disregarded here. The offset gets reset every time the HIP leaves the virtual object respecting the shifted boundaries.

$$\Delta O(n) = (e(n) - (e(n-1))) \cdot (\frac{k_{\mathrm{c}}}{k_{\mathrm{v}}} - 1) \tag{2.14a}$$

$$O(0) = 0 \tag{2.14b}$$

$$O(n) = \begin{cases} O(n-1) + \Delta O(n) & \text{if } e(n) > e(n-1) \wedge e(n-1) \leq e(n-2) \\ & \quad \wedge (k_{\mathrm{di}} < k_{\mathrm{de}} \wedge k_{\mathrm{di}} > 0) \\ O(n-1) - \Delta O(n) & \text{if } e(n) > e(n-1) \wedge e(n-1) \leq e(n-2) \\ & \quad \wedge (k_{\mathrm{di}} > k_{\mathrm{de}} \vee k_{\mathrm{di}} < 0) \\ 0 & \text{if } e(n) + O(n) < 0 \\ O(n-1) & \text{else} \end{cases} \tag{2.14c}$$

Depending on the sampling rate and the resolution of the haptic device, the offset will be updated frequently, triggered by even marginal movements. However, higher resolution and sampling rates will also lead to a decrease in the value of ΔO, so the steps responsible for the force augmentation would be smaller, which in turn leads to more of them being necessary to reach the equivalent to the desired stiffness. It might be beneficial to alter the way those step sizes are calculated in order to increase the effectiveness of the approach.

Of course, the additional offset could lead to difficulties as soon as the HIP leaves the original virtual object again, because at this point, the virtual spring will be attached to a point located outside of it. Therefore, there would be a discontinuity where the displayed force suddenly falls to zero once the original virtual object has been left. To avoid this, the force is continuously displayed until the offset surface of the virtual object is reached:

$$f(n) = \begin{cases} k_{\mathrm{v}} \cdot (e(n) + O(n)) & \text{if } e(n) + O(n) > 0 \\ 0 & \text{else} \end{cases} \tag{2.15a}$$

The rate-hardness achieved by the described approach is comparatively low because for the initial contact, the force will only increase very conservatively according to the inherently stable, but low stiffness k_{v}. In order to increase the rate-hardness for this approach, the haptic feedback forces f_{RH} are calculated directly according to the desired stiffness k_{de} for the initial contact of the HIP with the virtual object.

$$f_{\mathrm{RH}}(n) = k_{\mathrm{de}} \cdot e(n) \tag{2.16}$$

After the initial contact has been completed, i. e. when the HIP begins moving back out of the virtual object (n_{r}), the previously explained way of rendering the haptic feedback force will be applied. To ensure a smooth transition without any discontinuities in the displayed force, the additional deformation O of the spring is set to fit the force that is displayed at the time of the transition:

$$O(n_{\mathrm{r}}) = \frac{f_{\mathrm{RH}}(n_{\mathrm{r}})}{k_{\mathrm{v}}} - e(n_{\mathrm{r}}) \tag{2.17}$$

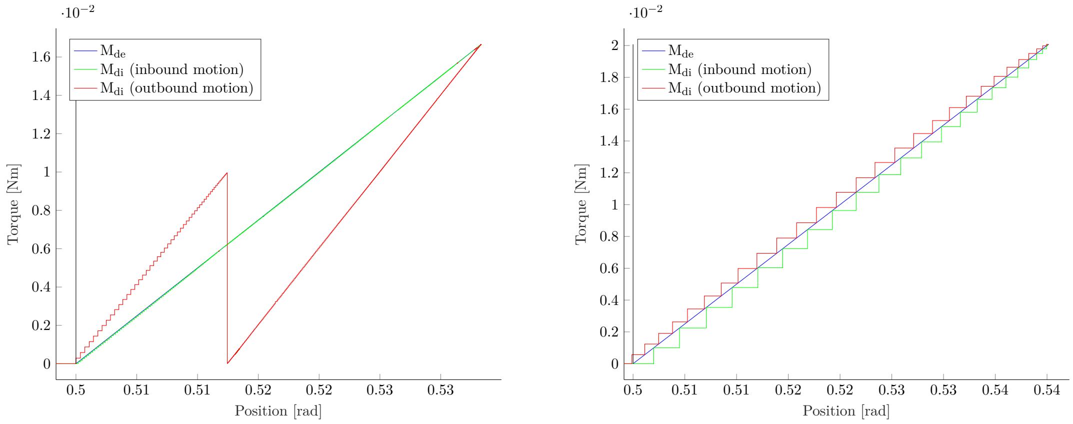
Chapter 3

Enhancements to the successive force augmentation approach

While the SFA approach provides many benefits over other haptic force rendering approaches, there are still some points that can be improved.

3.1 Transparently displaying low stiffness

In case the stiffness to display is very soft ($k_{\text{de}} < k_{\text{v}}$), the virtual spring's stiffness would be reached with SFA using a negative artificial offset $O < 0$ eventually. However, this negative offset would also result in the HIP leaving the virtual object while still being inside its original boundaries. The virtual offset gets reset at this point, shifting the object boundaries back to their original position. This however results in suddenly appearing forces as the HIP now is again inside the virtual object's boundaries:



(a) The original SFA approach displaying a stiffness of $k_{\text{de}} = 5\frac{Nm}{rad}$. Once the HIP leaves the virtual object's shifted boundaries, they get reset to their original position, resulting in more forces.

(b) A virtual spring without any additional controllers with a stiffness of $k_{\text{de}} = 5\frac{Nm}{rad}$ with no perceivable discontinuities.

Figure 3.1: For small desired stiffnesses, it is reasonable to disable the SFA controller.

Directly assuming a virtual spring of that exact stiffness instead of using SFA would result in greater transparency and is stable without an additional controller according to (1.8), so this is what will be done.

3.2 Stably delivering appropriate rate-hardness

For motions directed out of the virtual object, the stiffness used to compute the according force will always be k_{v}. Considering that for the initial contact, the used stiffness will be k_{de}, the total energy of the system could very easily become negative, so the passivity condition would be violated.

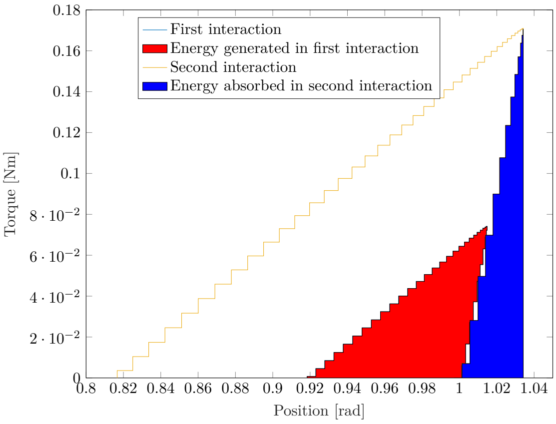


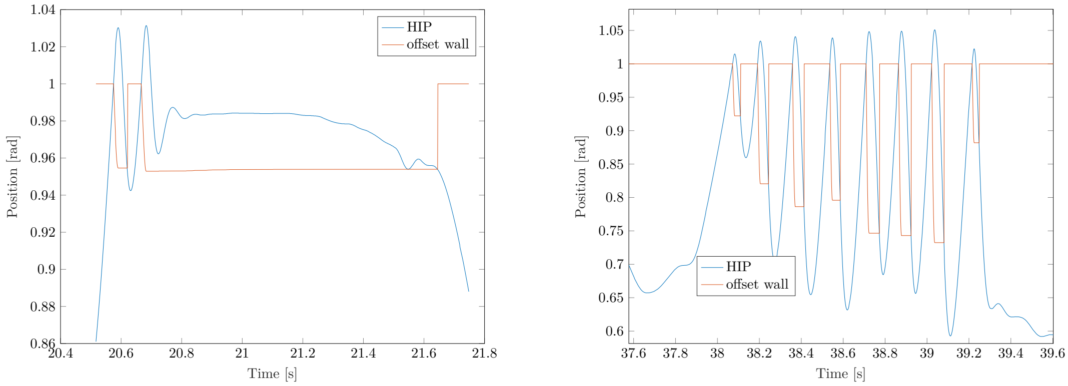
Figure 3.2: The SFA displaying a stiffness of $k_{\mathrm{de}} = 5\frac{Nm}{rad}$ with $k_{\mathrm{v}} = 0.8\frac{Nm}{rad}$. Even though some of the energy generated in the first interaction ($E_1 = -0.00310J$) has been dissipated externally before the second interaction ($E_{2,\ \mathrm{absorbed}} = 0.00264J$), it will generate even more energy. The system is unstable.

This can be shown in the position-force diagram Fig. 3.2, where in a single interaction cycle, the outbound motion will usually cover a significantly larger area underneath it than the corresponding motion into the virtual object. This results from k_{de} generally being much greater than k_{v}, so the potential energy stored in the stiffer spring will be considerably less than in a less stiff virtual spring, as shown in (3.1a). This equation compares the energy stored in the two springs at the maximum penetration depth (x_{m}). The stiffer spring (k_{de}) is connected to the original virtual surface x_{s}, whereas the spring with the lower stiffness requires more elongation to display the same force, so it is connected to the offset surface x_{os}.

$$\frac{1}{2} \cdot k_{\mathrm{v}} \cdot (x_{\mathrm{m}} - x_{\mathrm{os}})^2 \geq \frac{1}{2} \cdot k_{\mathrm{de}} \cdot (x_{\mathrm{m}} - x_{\mathrm{s}})^2 \tag{3.1a}$$

$$\frac{1}{2} \cdot \frac{{k_{\mathrm{de}}}^2}{k_{\mathrm{v}}} \cdot (x_{\mathrm{m}} - x_{\mathrm{s}})^2 \geq \frac{1}{2} \cdot k_{\mathrm{de}} \cdot (x_{\mathrm{m}} - x_{\mathrm{s}})^2 \quad \forall \quad k_{\mathrm{de}} \geq k_{\mathrm{v}} \tag{3.1b}$$

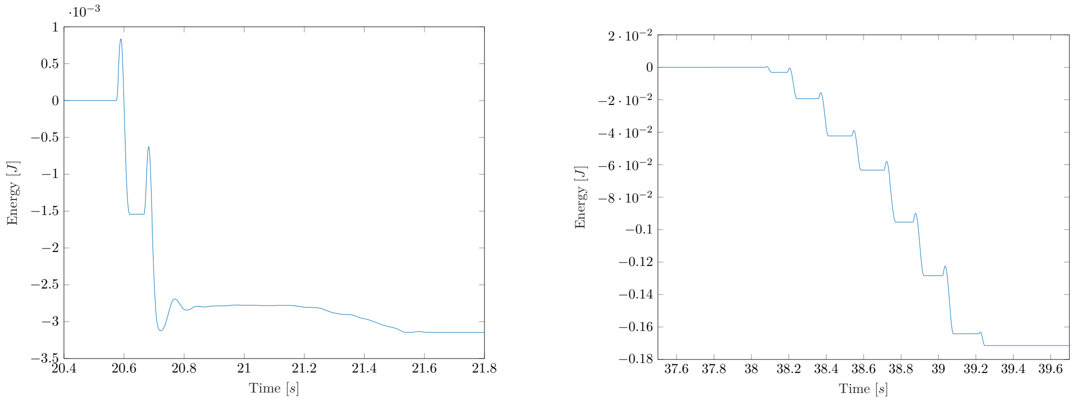
These effects get more relevant with increasing stiffness (see Fig. 3.3) as soon as the generated energy is sufficient to push the HIP out of the virtual object including the virtual offset. Then, the energy generated from the previous interaction will be further multiplicated in the next interaction (see Fig. 3.2).



(a) The SFA displaying a stiffness of $k_{\text{de}} = 2\frac{Nm}{rad}$. Minor oscillations can be identified, but the user's hand provides sufficient damping to retain stability.

(b) The SFA displaying a stiffness of $k_{\text{de}} = 5\frac{Nm}{rad}$. The resulting oscillations can not be damped out anymore, but their amplitude increases. The interaction is unstable.

Figure 3.3: Comparison of different stiffnesses displayed with SFA using the enhanced rate-hardness.



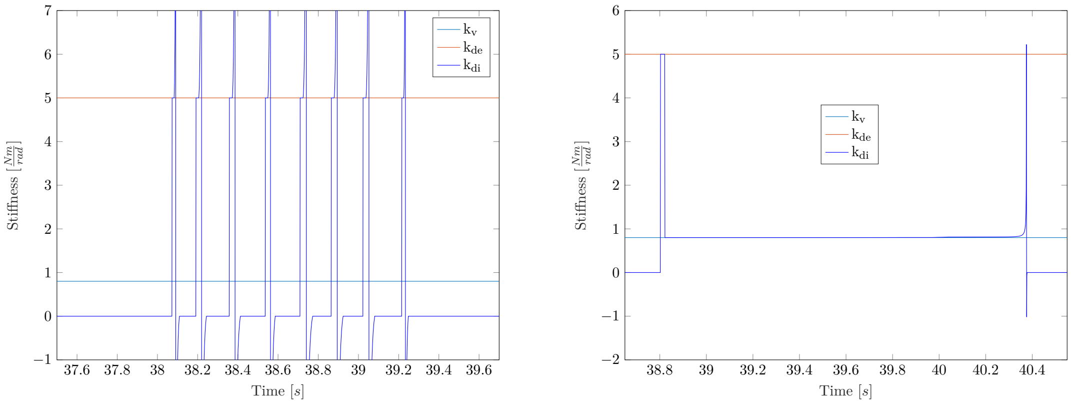
(a) The energy calculated from the interaction at $k_{\text{de}} = 2\frac{Nm}{rad}$. The energy quickly becomes negative, so the device is active. However, the operator's hand passively removes this energy through damping, so the interaction is stable.

(b) The energy at $k_{\text{de}} = 5\frac{Nm}{rad}$. The device generates sufficient energy to maintain oscillations regardless of external damping.

Figure 3.4: Comparison of the energy generated with the SFA approach using the enhanced rate-hardness. Although the device is active in both experiments, the interaction becomes unstable only for the higher stiffness.

One way to mitigate this unpleasant effect is to omit the adaption of the offset after the initial contact as it has been shown in (2.17) and condone with the resulting discontinuity in force. In most real implementations, the initial contact ends close to the virtual object's surface anyways because the displayed force increments very quickly according to k_{de}, so the difference between this displayed force on the initial contact and the following smaller force corresponding to k_{v} should not be noticeable.

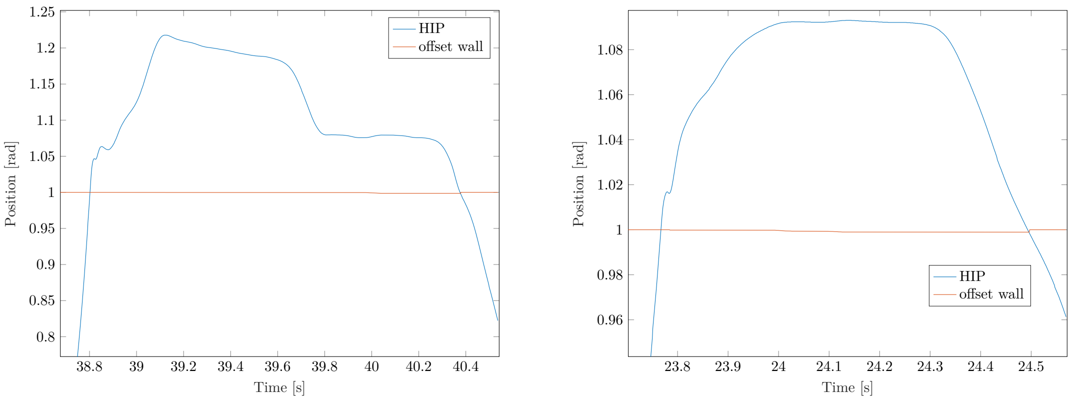
As Fig. 3.6 shows, the proposed solution works in keeping the contact stable as compared to the original approach. However, the displayed stiffness quickly drops to a low value as Fig. 3.5 reveals.



(a) At the beginning of every single contact cycle, the desired stiffness of $k_{\text{de}} = 5\frac{Nm}{rad}$ is applied and as soon as the HIP starts moving back out of the virtual object, the stiffness is changed to $k_{\text{v}} = 0.8\frac{Nm}{rad}$ while maintaining the force and accordingly changing the offset. This results in the displayed stiffness increasing in value considerably as the penetration depth decreases disproportionately quicker than the force.

(b) The modified approach does not maintain the force displayed when the HIP's movement changes direction after the initial contact. Instead, the offset is kept at zero until that point. Therefore, the stiffness drops down to k_{v}.

Figure 3.5: Comparison of the resulting stiffnesses with and without discontinuity in the force.



(a) The modified SFA displaying a stiffness of $k_{\text{de}} = 5\frac{Nm}{rad}$. No oscillations result from the initial contact.

(b) The modified SFA displaying a stiffness of $k_{\text{de}} = 50\frac{Nm}{rad}$. Even for extremely high stiffness, the initial contact does not lead to instability.

Figure 3.6: Comparison of different stiffnesses displayed with SFA using the enhanced rate-hardness with the initially fixed offset.

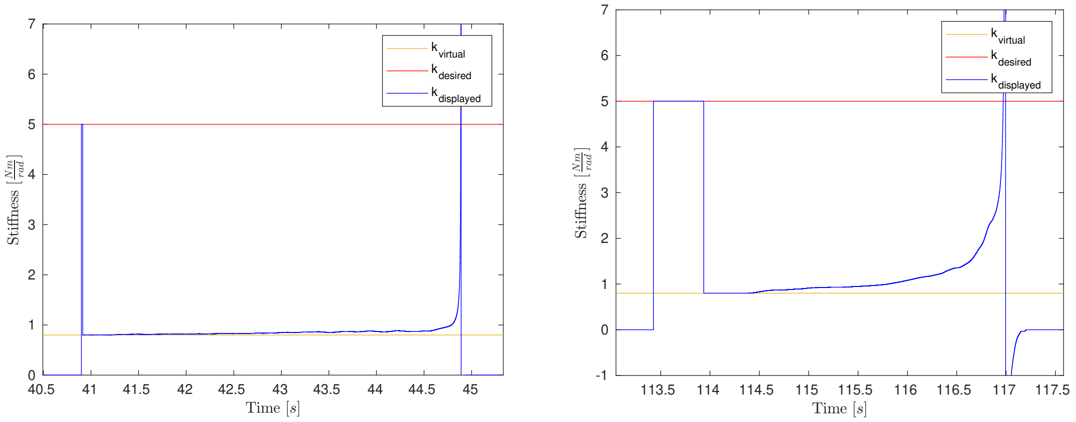
3.3 Accelerating the force augmentation

After the initial contact, it is desired to increase the virtual offset quickly in order to reach the desired stiffness without much delay. In order to reach the required offset as quickly as possible, it can be updated not only on the first sample of each interaction cycle, but instead on every single sample of each inbound motion:

$$O(n) = \begin{cases} O(n-1) + \Delta O(n) & \text{if } e(n) > e(n-1) \wedge (k_{\text{di}} < k_{\text{de}} \wedge k_{\text{di}} > 0) \\ O(n-1) - \Delta O(n) & \text{if } e(n) > e(n-1) \wedge (k_{\text{di}} > k_{\text{de}} \wedge k_{\text{di}} < 0) \\ 0 & \text{if } e(n) + O(n) < 0 \\ O(n-1) & \text{else} \end{cases} \tag{3.2}$$

This results in the force following a steeper curve with respect to the position. In particular, the displayed stiffness for the inbound motion is exactly equivalent to k_{c} as long as $k_{\text{di}} < k_{\text{de}}$:

$$\begin{aligned} \Delta O(n) &= (e(n) - e(n-1)) \cdot (\frac{k_{\text{c}}}{k_{\text{v}}} - 1) && (3.3a) \\ O(n) &= O(n-1) + \Delta O(n) && (3.3b) \\ f(n) &= k_{\text{v}} \cdot (e(n) + O(n)) && (3.3c) \\ &= f(n-1) + (e(n) - e(n-1)) \cdot k_{\text{c}} && (3.3d) \\ \Delta f &= \Delta e \cdot k_{\text{c}} && (3.3e) \end{aligned}$$



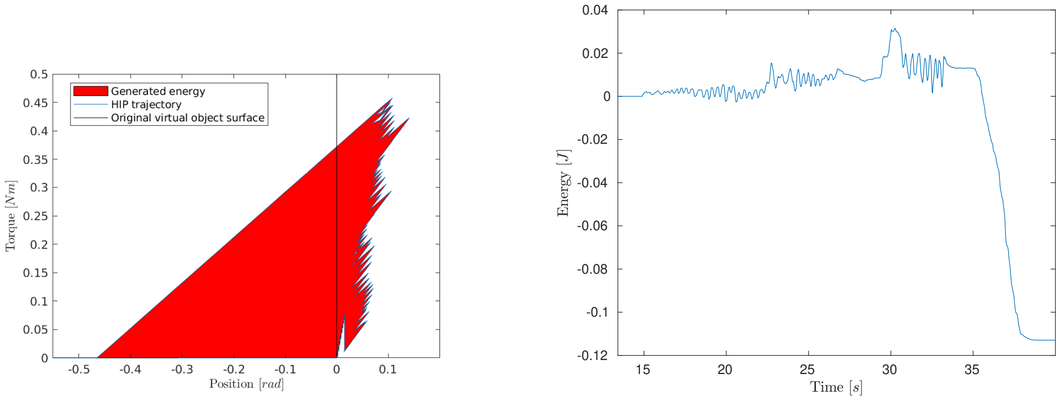
(a) The displayed stiffness over time with only a single offset update whenever the motion direction changes from outbound to inbound.

(b) The displayed stiffness over time with continuous offset updating.

Figure 3.7: Comparison of continuous offset update with the traditional method

3.4 Ensuring passivity

The SFA, even with the modified initial display of the correct rate-hardness (§ 3.2), can become active. This can be shown very clearly in the position-force diagram:



(a) The displayed force is successfully successively augmented. As a result, a considerable amount of energy is generated within the interaction.

(b) While the force is augmented, the energy does not change significantly. However, the energy stored in the virtual spring connected to the offset object surface increases. When the user moves back out of the object, all of that energy is released, making the device active.

Figure 3.8: Example of the SFA being active at $k_{\text{de}} = 5\frac{Nm}{rad}$

In order to solve that problem, the TDPA is implemented. All the forces resulting from the SFA approach are observed and regulated by its passivity controller before being displayed. This way, any excess energy generated in an interaction gets dissipated by the adaptive damping of the TDPA instead of contributing to instability.

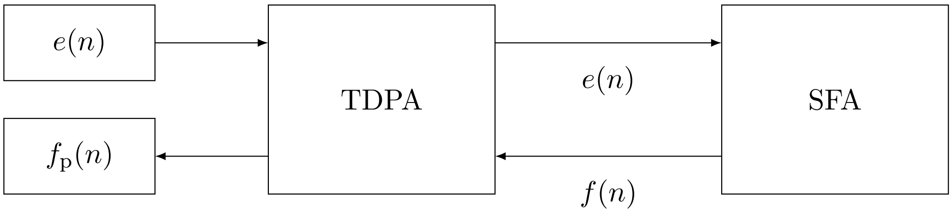
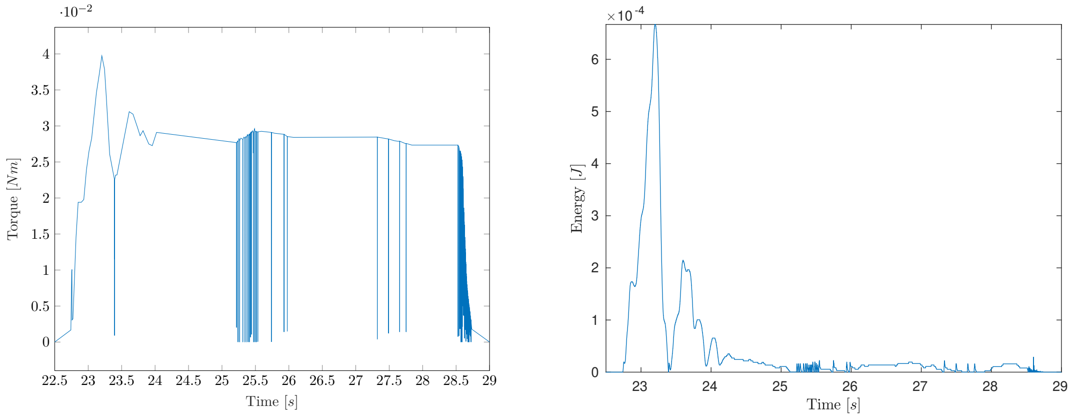


Figure 3.9: The forces computed with SFA are ensured to result in a passive interaction (with a passive force f_{p}) through the serial addition of a TDPA implementation.

3.5 Passively filtering discontinuous forces

As explained in § 2.1 and illustrated in Fig. 3.10, the passivity controller can lead to rather extreme discontinuities in force due to the aggressive adaptive damping.

One approach aiming to avoid those problems is to implement a filter that decouples the HIP from the direct feedback from the passivity controller. Thereby, the inconsistent forces are only applied on a virtual mass before being transferred to the HIP. In theory,



(a) The torque over time with adaptive damping from a passivity controller.

(b) The energy computed by the passivity observer.

Figure 3.10: Undesirable discontinuities in the torque resulting from the passivity controller. Whenever the observed energy reaches zero, the displayed torque gets suddenly altered by a perceivable margin.

it is possible to create a coupling that effectively minimizes the effects of the oscillating forces in a way that oscillations are not perceivable at the HIP.

The filter used here and illustrated in Fig. 3.11 is comprised of a virtual mass that is connected to the HIP via a virtual spring with the stiffness k_{f} and a virtual damper with the coefficient c_{f}. Additionally, it is connected to a fixed point with another virtual damper with the parameter c_{env}. This construction of the filter ensures the filter itself being passive. The entire control approach is then applied to its virtual mass instead of the HIP itself. As a result, the forces acting on the HIP are significantly smoother as shown in Fig. 3.12.

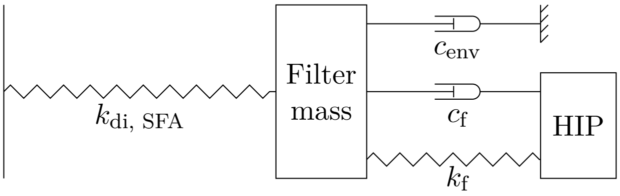


Figure 3.11: The virtual filter mass is connected through SFA to the boundary of the virtual object. The HIP is connected only to the filter mass, but not directly to the virtual object.

Unfortunately, this addition negatively affects transparency. For instance, some damping is introduced into the loop. Additionally, the value of the stiffness displayed at the HIP is lower than the desired stiffness, because the HIP is effectively coupled to the boundary of the virtual object only through serially interconnected springs instead of a single one as shown in Fig. 3.13:

$$k_{\mathrm{di,\,HIP}} = \frac{1}{\frac{1}{k_{\mathrm{di,\,SFA}}} + \frac{1}{k_{\mathrm{f}}}} \tag{3.4}$$

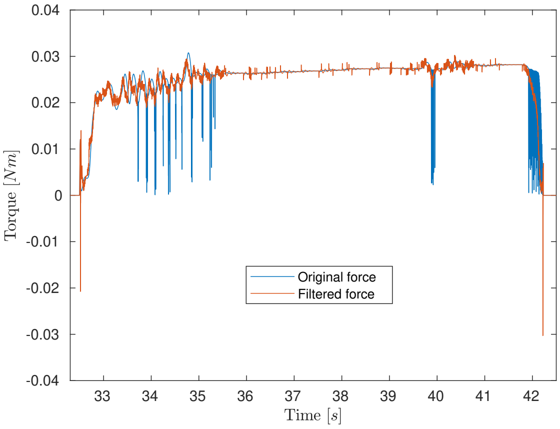


Figure 3.12: The filter successfully smooths out the discontinuities provided by the passivity controller.

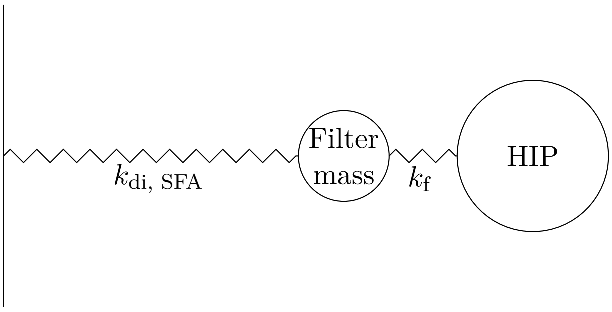


Figure 3.13: Regarding only stiffness, the interconnection between the HIP and the virtual object is equivalent to a serial interconnection of two springs.

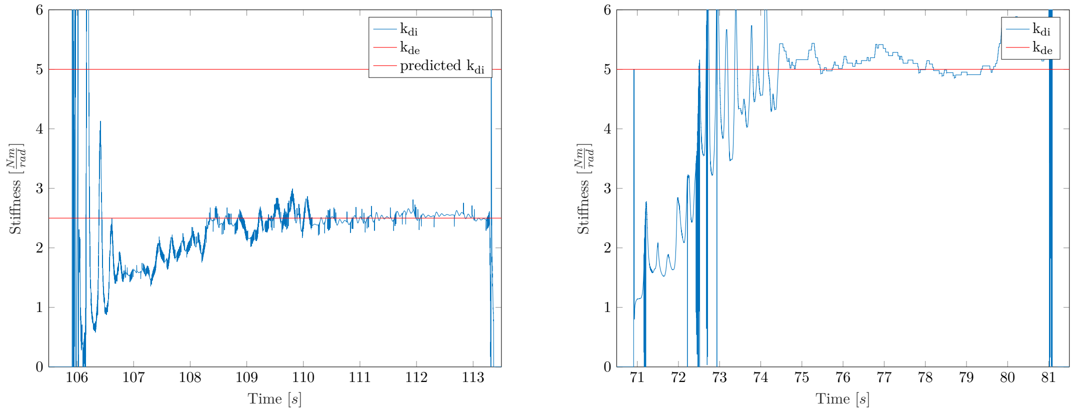
A very high stiffness k_{f} for the interconnection between the filter mass and the HIP is very useful to enhance transparency because it reinforces this coupling. However, it thereby decreases the filter's effect. In the extreme case, the oscillating forces produced by the passivity controller would be directly transferred to the HIP, so the filter misses its purpose. Furthermore, this stiffness is limited by the sampling frequency.

In order to provide better transparency regarding stiffness, the desired stiffness $k_{\text{de, SFA}}$ used in the SFA implementation could be calculated according to (3.4) to result in a desired perceived stiffness $k_{\text{de, HIP}}$ at the HIP. The difference is very clearly displayed in Fig. 3.14.

$$k_{\text{de, SFA}} = \frac{1}{\frac{1}{k_{\text{de, HIP}}} - \frac{1}{k_{\text{f}}}} \tag{3.5}$$

Note however that the resulting stiffness will only yield a meaningful (i. e. existing and positive) result for any $k_{\text{de, HIP}} < k_{\text{f}}$. This means that this implementation of a filter would result in additional limitations to the impedance range.

The SFA generally works by adjusting the virtual wall's position by introducing an artificial offset until the desired force for a given position is reached. In the filtered case, the SFA approach will base this desired force on the position of the virtual mass of the filter. As a result, the desired force calculated by the SFA generally leads to a lower



(a) The stiffness resulting from the serial interconnection of the filter and SFA.

(b) The stiffness using the proposed adapted value for $k_{\text{de, SFA}}$.

Figure 3.14: The effect of the serial interconnection of the virtual filter and the SFA on the achieved stiffness. The adapted value for $k_{\text{de, SFA}}$ ensures the correct stiffness to be reached.

perceived stiffness overall because the end effector is actually further inside the virtual object than the filter mass.

Feeding the stiffness that is really perceived by the user into the SFA implementation yields significantly better results. While the serial connection of the SFA and the filter can be interpreted as in Fig. 3.13 where the perceived stiffness is limited to $k_{\text{de, HIP}} < k_{\text{f}}$, considering the functioning of SFA leads to the following interpretation:

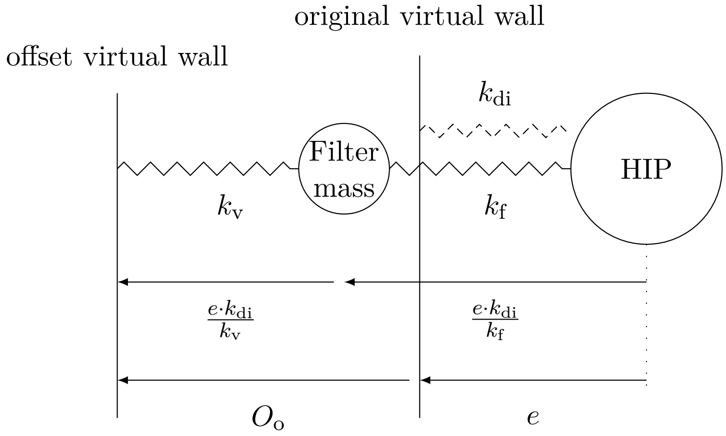


Figure 3.15: A more elaborate interpretation of the SFA and the filter in series. Note that the SFA includes the passivity controller at this point.

It is clear that this configuration is (at least in theory) not limited in the displayable stiffness. In fact, the optimal artificial offset (O_{o}) for any filter stiffness and position (of force) can be calculated:

$$O_{\text{o}} = e \cdot \left(\frac{k_{\text{de}}}{k_{\text{f}}} + \frac{k_{\text{de}}}{k_{\text{v}}} - 1\right) \tag{3.6}$$

Assuming $k_{\mathrm{de}} > k_{\mathrm{v}}$, which is justified because for smaller stiffnesses, the controller would be disabled anyways according to § 3.1, the offset will always be positive, so a phenomenon as described in § 3.1 is impossible. Fig. 3.16 shows how even a very high desired stiffness $k_{\mathrm{de}} = 5\frac{Nm}{rad}$ is asymptotically reached. It also shows how the desired rate-hardness is transparently achieved by disabling the filter for the initial contact.

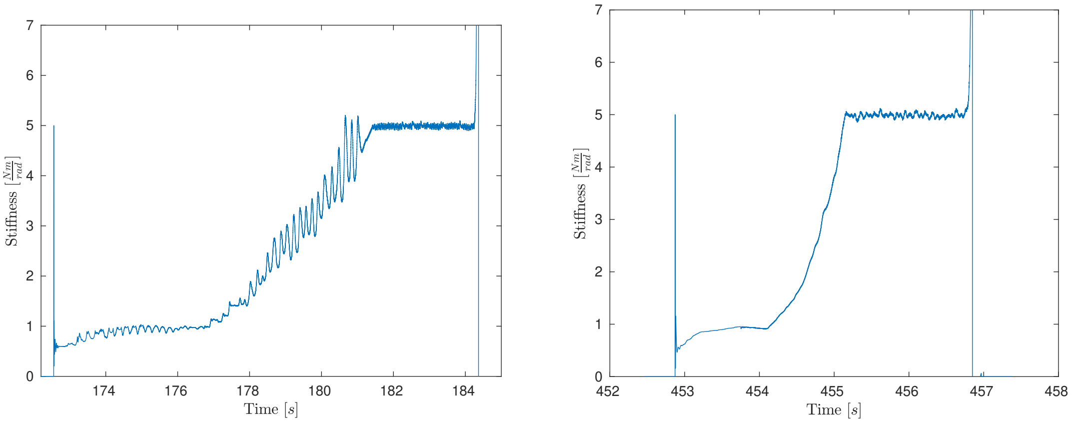
3.6 Alternative approach to control the offset steps

Fig. 3.7 shows progress in the response time for the SFA to reach the desired stiffness. However, there is still some potential for further improvements. A different way to control the steps in the artificial offset $O(n)$ is shown here. The ideal value ΔO_{i} which would result in the correct displayed force according to the current $e(n)$ and k_{de}, can be determined as follows:

$$\Delta O_{\mathrm{i}}(n) = |\frac{|k_{\mathrm{de}} \cdot e(n)| - |f(n)|}{k_{\mathrm{v}}}| \tag{3.7}$$

In order to avoid significant jumps in $O(n)$ which would result in equally as significant jumps in the displayed force and therefore would distort the user experience, it is useful to limit the value of $\Delta O(n)$. One meaningful way of doing this is by decreasing it to a fraction $c < 1$ of the proposed ideal value. This might also help with avoiding overshoot and an overall smoother experience.

$$\Delta O(n) = c \cdot \Delta O_{\mathrm{i}}(n) \tag{3.8}$$



(a) With the original step size, the filtered forces display the desired stiffness eventually.

(b) The adaptive step size enables the desired stiffness to be reached remarkably quicker.

Figure 3.16: The difference in response time for the different step sizes at a desired stiffness of $k_{\mathrm{de}} = 5\frac{Nm}{rad}$.

Another reasonable action is to constrain the step in the displayed force to a percentage $b < 1$ of it, according to Weber's Law, so no noticeable discontinuity in the force f_f occurs. This is done in the following way:

$$\Delta f_\mathrm{f}(n) = |b \cdot f(n-1)| \tag{3.9a}$$

$$\Delta O(n) \leq \frac{\Delta f_\mathrm{f}(n)}{k_\mathrm{v}} \tag{3.9b}$$

The resulting algorithm to successively augment displayed haptic forces proves to be remarkably quicker at doing so compared to the original approach as shown in Fig. 3.16:

3.7 Avoiding oscillations near the virtual wall

For high stiffness, the equilibrium point for a given force will be very close to the virtual object's original surface. However, in the vicinity of this original surface, the value of the displayed stiffness $k_\mathrm{displayed}$ is very sensitive to even small changes in the HIP's position:

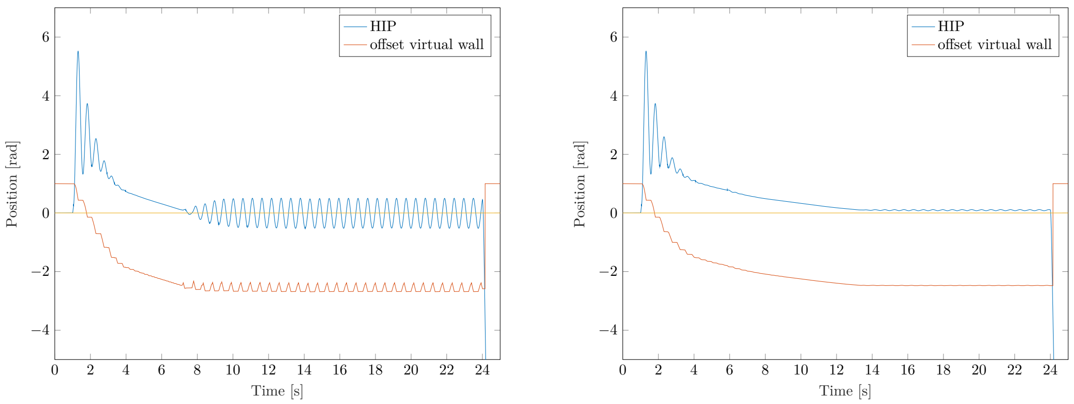
$$\Delta k_\mathrm{di} = \frac{f(n-1)}{e(n-1)} - \frac{f(n-2)}{e(n-2)} = \frac{f(n-1) \cdot e(n-2) + f(n-2) \cdot e(n-1)}{e(n-1) \cdot e(n-2)} \tag{3.10}$$

Therefore, the small changes in the virtual offset, which will lead to similarly small changes in the HIP's position result in significant changes in the value of k_di. Subsequently, the offset gets adjusted to again reach $k_\mathrm{di} = k_\mathrm{de}$. However, it is very likely to experience at least a small amount of overshoot due to system dynamics. Even though this overshoot might be only marginal, it will again have serious consequences in the displayed stiffness. This closed loop can lead to oscillations.

In order to counteract these oscillations, the offset step size can be further limited in the vicinity of the virtual surface to a fraction $a < 1$ of the distance to the original virtual wall. This way, the oscillations' amplitudes are largely reduced so they can't be felt anymore in the real implementation or even don't have any effect at all:

$$\Delta O(n) \leq a \cdot e(n); \tag{3.11}$$

However, it should be noted that this additional limitation of the step size can lead to an increased response time as Fig. 3.17 reveals.



(a) The HIP and the virtual surface positions show oscillations as the HIP reaches the vicinity of the virtual wall.

(b) The additionally limited values for ΔO result in reduced amplitude. However, the oscillations are not entirely removed.

Figure 3.17: The difference in oscillation amplitude, shown here in simulation results for better comparability.

3.8 Summary of modifications

The resulting enhanced SFA approach can be illustrated as a block diagram the following way:

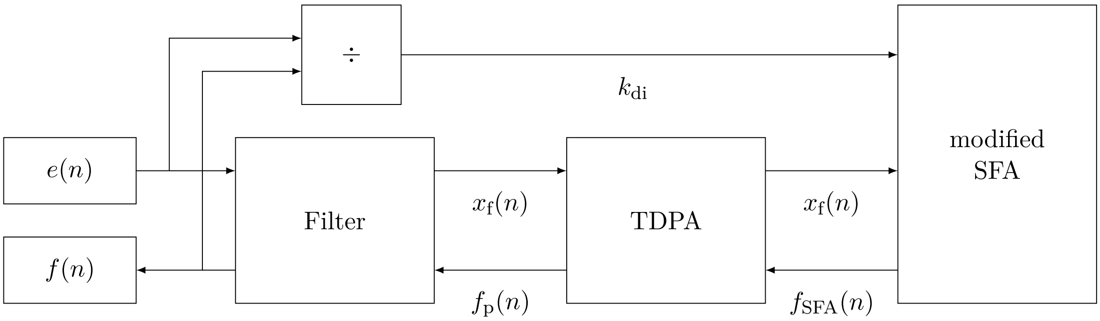


Figure 3.18: The block diagram for the enhanced SFA approach.

The diagram shows the penetration depth e being used to determine the currently displayed stiffness (k_{di}) as well as the position of the virtual filter mass, x_{f}. This position then passes through the TDPA implementation, where it is used to calculate the energy, before being forwarded into the SFA approach together with k_{di} to compute a force, f_{SFA}. This force is ensured to be passive by the TDPA, resulting in f_{p}. That force is then fed to the virtual filter, which outputs f, the kinaesthetic feedback force to be displayed by the haptic interface.

To show more detail of the exact process, the following (Algorithm 3.1) displays how the haptic feedback force f_{SFA} is determined from the modified SFA:

Algorithm 3.1: How $f_{\text{SFA}}(n)$ is determined inside the modified SFA

```
Result: $f(n)$
begin
    if $e(n) + O(n) > 0$ then
        if inbound motion then
            if initial contact then
                $f_{\text{SFA}}(n) = -k_{\text{de}} \cdot e(n)$;
            else
                $\Delta O(n) = c \cdot (|\frac{|k_{\text{de}} \cdot e(n)| - |f(n-1)|}{k_{\text{v}}}|)$;
                $\Delta O(n) = min(\Delta O(n), b \cdot \frac{f(n-1)}{k_{\text{v}}})$;
                $\Delta O(n) = min(\Delta O(n), a \cdot e(n))$;
                if $k_{di} < k_{de}$ then
                    $O(n) = O(n-1) + \Delta O(n)$
                else
                    $O(n) = O(n-1) - \Delta O(n)$
                $f_{\text{SFA}}(n) = -k_{\text{v}} \cdot (e(n) + O(n))$;
        else
            $f_{\text{SFA}}(n) = -k_{\text{v}} \cdot (e(n) + O(n))$;
    else
        $f_{\text{SFA}}(n) = 0$;
        $O(n) = 0$;
```

Chapter 4

Extension of the successive force augmentation approach to three degrees of freedom

The approaches discussed so far have been described in a single dimension ($f_1(x)$) as this is a more simple way to effectively develop them and finally show how they work. However, for real implementation it is quite useful to have multiple DoF ($\boldsymbol{f}_\mathrm{n}(\boldsymbol{x})$), ideally up to six to account for all possible combinations of rotation and translation. For many applications though, three DoF in translation are sufficient.

A typical input for these functions is the penetration vector $\boldsymbol{e}(n)$.

There are two manifest approaches to extending those approaches to multiple DoF, vectorial concatenation and the application of the approach on the magnitude of the displayed force.

4.1 Vectorial concatenation

A typical way to advance from one to multiple DoF is to simply build a vector consisting of multiple instances of the single-DoF-approach:

$$\boldsymbol{f_n}(\boldsymbol{x}) = \begin{bmatrix} f_1(x_1) \\ f_1(x_2) \\ \vdots \\ f_1(x_\mathrm{n}) \end{bmatrix} \tag{4.1}$$

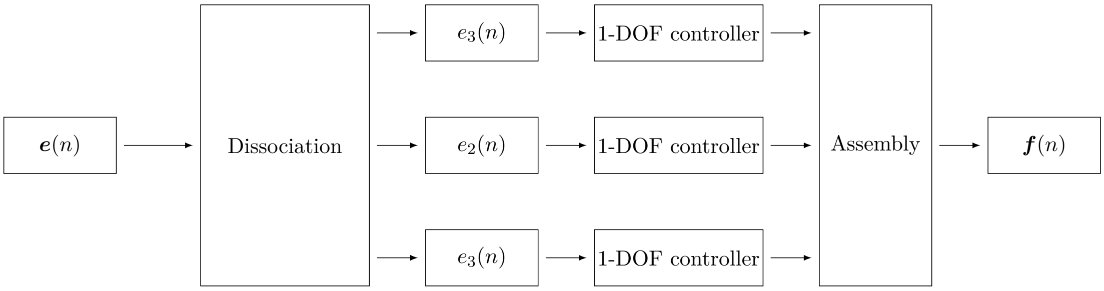


Figure 4.1: Block diagram of the vectorial concatenation

4.2 Application on magnitude

A different idea to extend a haptic rendering to multiple DoF is to disassemble the input vector into its direction $\boldsymbol{u}$ and its magnitude x_{m}. Then the result can be computed separately. The resulting magnitude is determined using the single-DoF-approach, whereas the direction $\boldsymbol{u}_{\mathrm{f}}$ has to be determined in an appropriate way depending on $\boldsymbol{u}$:

$$x_{\mathrm{m}} = norm(\boldsymbol{x}) = \sqrt{\sum_{i=0}^{n} x_i^2} \tag{4.2a}$$

$$\boldsymbol{u} = \frac{\boldsymbol{x}}{x_{\mathrm{m}}} \tag{4.2b}$$

$$\boldsymbol{f}_{\mathrm{n}}(\boldsymbol{x}) = f_1(x_{\mathrm{m}}) \cdot \boldsymbol{u}_{\mathrm{f}}(\boldsymbol{u}) \tag{4.2c}$$

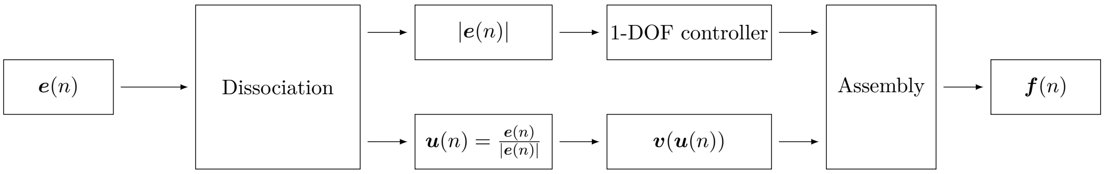


Figure 4.2: Block diagram of the application on magnitude

Often, the haptic force's direction is meant to point outside the virtual object and rather easy to calculate. This is especially true if the input represents the vector from the closest point on the virtual object's surface to the HIP:

$$\boldsymbol{u}_{\mathrm{f}}(\boldsymbol{u}) = -\boldsymbol{u} \tag{4.3}$$

In some cases, the application on the force's magnitude and the vectorial concatenation actually yield identical results ($\boldsymbol{F}_\mathrm{M} = \boldsymbol{F}_\mathrm{V}$). An example for this is a simple spring without any additional controller:

$$\boldsymbol{F}_\mathrm{M}(n) = |\boldsymbol{e}(n)| \cdot k \cdot (-\boldsymbol{u}(n)) \tag{4.4a}$$

$$\boldsymbol{F}_\mathrm{V}(n) = \begin{bmatrix} k \cdot -e_1(n) \\ k \cdot -e_2(n) \\ k \cdot -e_3(n) \end{bmatrix} \tag{4.4b}$$

$$|\boldsymbol{e}(n)| \cdot k \cdot (-\frac{\boldsymbol{e}(n)}{|\boldsymbol{e}(n)|}) = k \cdot -\boldsymbol{e}(n) = \begin{bmatrix} k \cdot -e_1(n) \\ k \cdot -e_2(n) \\ k \cdot -e_3(n) \end{bmatrix} \tag{4.4c}$$

However, many approaches contain some instances of memory, such as accumulated energy or the additional offset for SFA. These can lead to remarkable differences between the two approaches on multidimensional implementation.

4.3 Determining the penetration depth

In contrast to a 1-DoF device, a multi-DoF system can display complex multidimensional shapes and objects. This leads to the question on how the crucial variable $e(n)$ is to be computed as it is not always obvious.

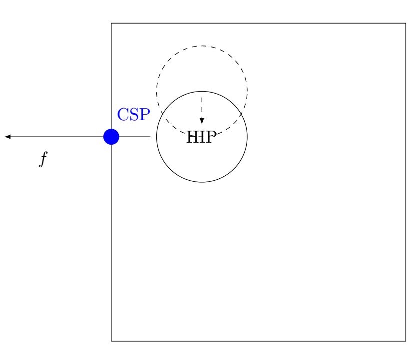
4.3.1 Closest surface point

One of the self-evident solutions to this problem is to determine the point on the virtual object's surface which is closest to the HIP's current position (x_{CSP}) and to direct all the forces there. Then, $\boldsymbol{e}(n)$ is calculated the following way:

$$\boldsymbol{e}(n) = x(n) - x_{CSP}(n) \tag{4.5}$$

However, especially for more complex virtual objects, it might be difficult to determine this closest surface point (CSP) in the first place.

Apart from the difficulties in calculating the CSP which could be solved by efficient algorithms, the force computed using this approach might also feel very odd in some cases. Depending on the virtual object's shape, there could be sudden jumps in the CSP, resulting in corresponding jumps of the haptic force's direction, as shown in Fig. 4.3. Even though the magnitude should be continuous, the user experience would be significantly distorted.



(a) The virtual object has been entered from above, and the rendered haptic force is directed as expected

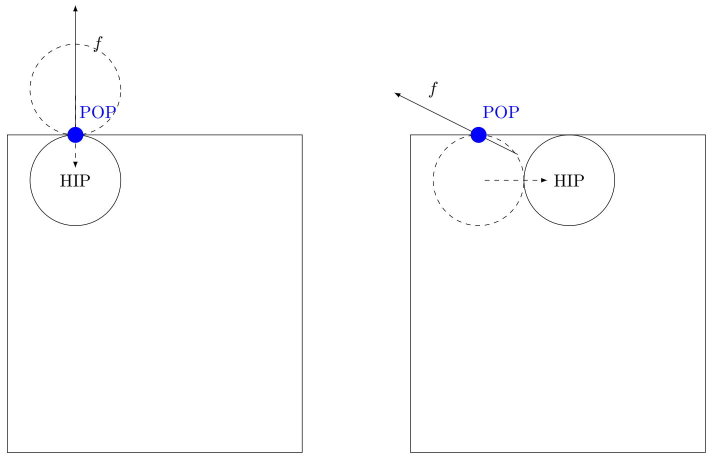
(b) After some further movement, the rendered haptic force is suddenly directed to an unexpected direction

Figure 4.3: Jumping CSP and haptic force

4.3.2 Point of penetration

The explained jumps in the displayed force can be avoided using an alternative method which is based on recording the exact point where the HIP initially entered the virtual object (x_{POP}). Then, $\boldsymbol{e}(n)$ is always directed at this point:

$$\boldsymbol{e}(n) = x(n) - x_{POP} \tag{4.6}$$



(a) The virtual object has been entered from above, and the rendered haptic force is directed as expected

(b) After some lateral movement, the rendered haptic force is directed to an unexpected direction

Figure 4.4: Unexpectedly directed haptic force with POP

It becomes apparent quite quickly that this method distorts the experience as well, as Fig. 4.4 illustrates. For instance, if the user tries to move the HIP across a virtual surface in order to explore the object, it would be pulled back to the POP where the HIP first made contact with the surface, which would for example lead to the perception of a concave surface instead of a flat one.

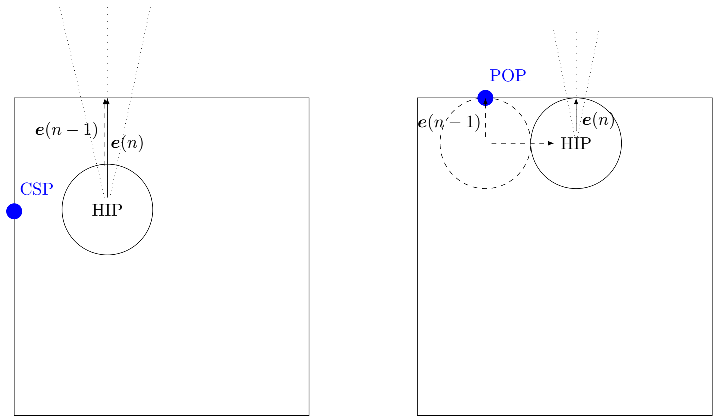
Another problem of this approach is dealing with the HIP leaving the virtual object at a different point than the POP. For perfect transparency, the force would need to be set to zero, however, this would lead to a perceptible discontinuity.

4.3.3 Restricted surface point

One way of overcoming the previously described difficulties is an approach that combines the explained methods to find $\boldsymbol{e}(n)$. The main idea is to define a point on the object's surface that is not fixed in place. It is similar to the CSP, but restricted in its lateral movement.

One way of achieving this behavior is to ensure that the movement of the point on the surface in between samples is restricted to a reasonable distance. This can be achieved by defining an infinitely long right circular cone with its rotation axis parallel to $\boldsymbol{e}(n-1)$ and its apex on the current HIP. Then, $\boldsymbol{e}(n)$ is found as the CSP contained in the cone. Thus, this method can be interpreted as an additional limitation to the CSP. Effectively, the angle between two subsequent values of $e(n)$ can be limited according to the ratio chosen for the cone. This ratio serves as the ratio of perpendicular to parallel force as compared to the previous direction.

The previously explained problems occurring with CSP and POP can be solved using this approach as shown in Fig. 4.5.



(a) Although the CSP is to the side of the HIP, the force still keeps its previous direction.

(b) Regardless of lateral movement, the rendered haptic force is directed in the expected direction

Figure 4.5: More intuitive $\boldsymbol{e}$ computation with the proposed approach. The dotted lines represent the cone.

4.4 Vectorial concatenation of the successive force augmentation

When applying the SFA on the individual coordinates of the haptic force vector, this vector itself needs to be determined first. One of the previously explained approaches, such as CSP, could be used to do that. Then, the force can be computed according to § 4.1.

However, it is important to consider the changes of the virtual object's boundaries during the interaction. Those boundaries will be moved out of the virtual object to increase the forces in order to display a high stiffness. That means that there might be forces to display even if the HIP is not inside the original virtual object, but only inside its artificially extended boundaries.

To properly obtain these forces, it is useful to compute the CSP and thereby if the HIP is inside the original virtual object (determined by the boolean value b_i) at all times, even when this is not the case. Then, before feeding the individual coordinates of the value obtained for $\boldsymbol{e}_\mathrm{env}$ from the virtual environment into different instances of the SFA approach, they get further processed. To properly leverage the effectiveness of SFA, their absolute value will be determined and assigned a sign depending on the HIP's position. If it is outside the original virtual object, the individual coordinates will each get a minus sign, else they will all get a positive one. Thereby, the SFA implementation as explained in § 3 will interpret those values accordingly as inside or outside the original boundaries and adjust those accordingly.

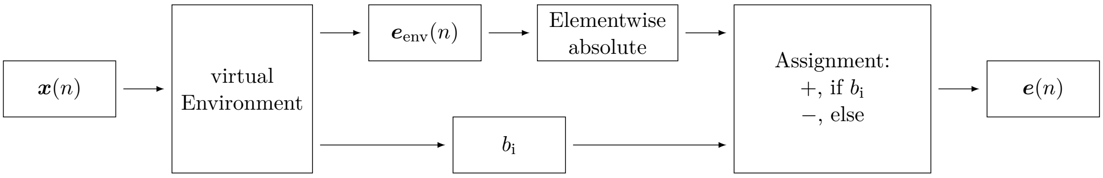


Figure 4.6: The calculation of $e(n)$ for the vectorial concatenation

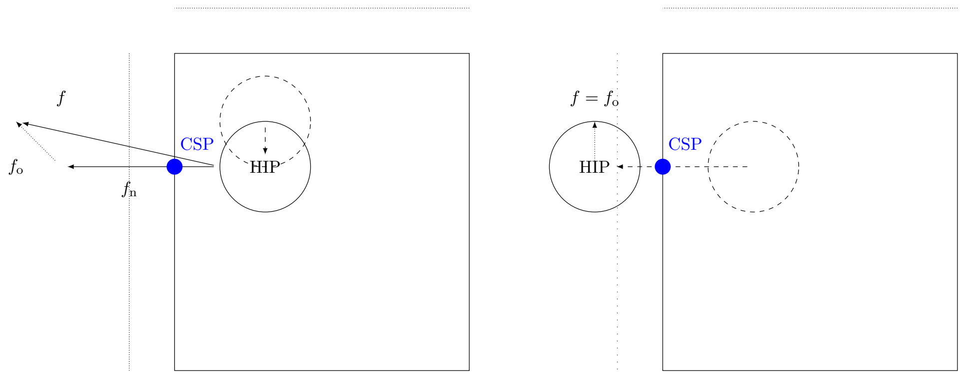
Of course, this manipulation of $\boldsymbol{e}_\mathrm{env}(n)$ will only yield negative forces as results. The individual coordinates need to be adjusted before being displayed in a way that they oppose the respective coordinate of $\boldsymbol{e}_\mathrm{env}(n)$.

$$f_\mathrm{i}(n) = \mathrm{sgn}(e_{\mathrm{env,\ i}}(n)) \cdot |f_{\mathrm{SFA,\ i}}(n)| \tag{4.7}$$

4.4.1 Resetting the memory after the haptic interaction point has left a virtual object

There might be forces to display outside of the virtual object depending on the virtual offsets to the object boundaries. The decision if the HIP has left the object is comparatively complex in the three-dimensional implementation.

It might happen that according to one of the dimensions, the HIP has left an virtual object while another dimension suggests that it is still inside. One method of dealing with this kind of contradicting data is to simply leave all the individual instances of SFA completely decoupled. However, this could lead to situations where forces would be displayed even after a virtual object has been left even regarding the offset boundaries.



(a) The CSP has just changed. As a result, the forces computed directly based on e have changed, but the additional offset remains.

(b) After some lateral movement, there is no force due to e anymore, but the vertical offset has not been overcome yet, so there is still a force displayed.

Figure 4.7: Remaining force due to virtual offset. f_o is the portion of the force generated from the artificial offset, whereas f_n is the part depending solely on the e itself.

Of course, the offset would be updated as the interaction continues (which is in this case even after the virtual object has been left horizontally). The displayed stiffness in the vertical direction is not computable because of a division by zero (vertical penetration depth $e_\mathrm{v} = 0$). In case this fraction is then interpreted as ∞, the offset would be decreased over the course of the interaction according to (3.2).

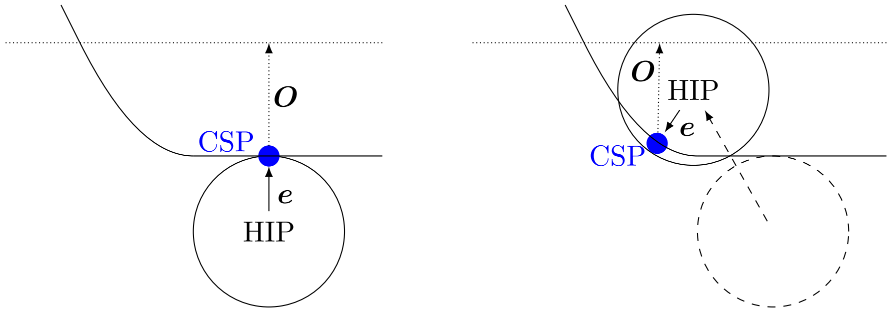
However, in the case of a vertical wall as displayed in Fig. 4.7, the condition $e_\mathrm{v}(n) > e_\mathrm{v}(n-1)$ would not be met as long as the CSP is on that vertical wall because $e_\mathrm{v}(n) = e_\mathrm{v}(n-1) = 0$. That means that in order to reset the vertical force caused by the additional offset, the HIP needs to be moved to a position where $e_\mathrm{v} \neq 0$. If that point is outside the shifted boundaries of the virtual object, the force would suddenly jump because the offset would get reset.

Apart from these undesirable forces, the energy computed for the passivity controller would not be reset either until the HIP leaves the offset virtual object.

In order to avoid this behavior, it makes sense to couple the individual coordinates in a way that their memory can be reset after a virtual object has been left.

One rather straightforward approach to determine if the object has been left is to simply compare the offset values with their respective corresponding coordinate of $\boldsymbol{e}$ individually. As long as the HIP is outside of the virtual object, this works well because the offset will be zero in all coordinates. As soon as some offset has been built to display a high stiffness, there are different ways to decide if the HIP is inside the virtual object.

One approach is to reset the controller once the object has been left according to at least one coordinate. For the previous example, this works well. As soon as the virtual object has been left horizontally, considering the offset boundary, the offset for every single coordinate would be reset, so there would be no undesired values in force. However, this condition is not sufficient for some cases.



(a) Some considerable vertical offset has been generated to display a high stiffness.

(b) After some lateral movement, e has slightly changed: $e_{\text{horizontal}} \neq 0$. However, the horizontal offset has not accumulated yet. Therefore, the horizontal coordinate reports that the virtual object has been left ($e_{\text{h}} + O_{\text{h}} < 0$), and the vertical force suddenly drops because its offset has been reset.

Figure 4.8: Discontinuities with single coordinate triggered resetting

Unfortunately, requiring all coordinates to each individually leave the virtual environment before activating a global reset is not an option either as its results include the difficulties yielded from the entirely decoupled approach.

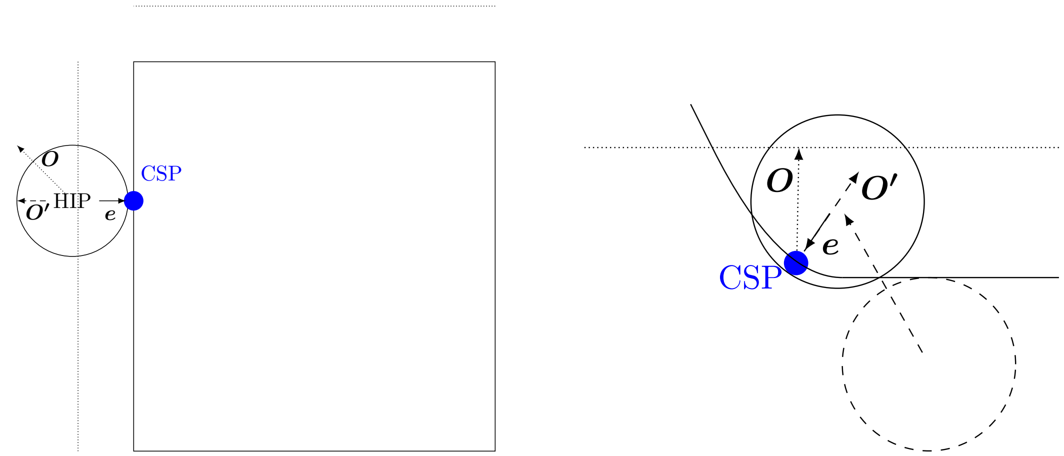
A satisfying solution is to consider the direction of $\boldsymbol{e}$ to determine to what extent the offset is relevant in compensating the distance from the virtual object. This is done by computing the projection of $\boldsymbol{O}$ on $\boldsymbol{e}$, called $\boldsymbol{O'}$. Then, it can be determined if that projection is sufficient to keep the HIP inside the shifted boundaries ($b_{\text{i}} = true$)of the virtual object or if it has actually left the object entirely ($b_{\text{i}} = false$):

$$\boldsymbol{O'} = \frac{\boldsymbol{e}(n-1) \cdot \boldsymbol{O}(n-1)}{\boldsymbol{e}(n-1) \cdot \boldsymbol{e}(n-1)} \cdot \boldsymbol{e}(n-1) \tag{4.8a}$$

$$\text{solve } \boldsymbol{e}(n-1) = \alpha \cdot \boldsymbol{O'} \quad \text{for } \alpha \tag{4.8b}$$

$$b_{\text{i}}(n) = \begin{cases} true & \text{if } \alpha < 1 \\ false & \text{else} \end{cases} \tag{4.8c}$$

All of the previous examples of difficult phenomena appearing with different approaches to the definition of the object boundaries are shown in Fig. 4.9 to work well with the proposed solution.



(a) As soon as $|\boldsymbol{e}| > |\boldsymbol{O'}|$, where $\boldsymbol{O'}$ is the projection of $\boldsymbol{O}$ onto $\boldsymbol{e}$, and they are directed opposingly, the condition for the global reset is met.

(b) $|\boldsymbol{O'}| > |\boldsymbol{e}|$ and they are directed opposingly, so the HIP is determined to still be inside the virtual object.

Figure 4.9: Vector projection-based reset condition

4.5 Application of successive force augmentation on magnitude

The application of the SFA approach on the magnitude of the displayed force is less complex than the vectorial concatenation. A similar way to pre-process $\boldsymbol{e}_{\text{initial}}(n)$ is applied for the same reasons as explained in § 4.4.

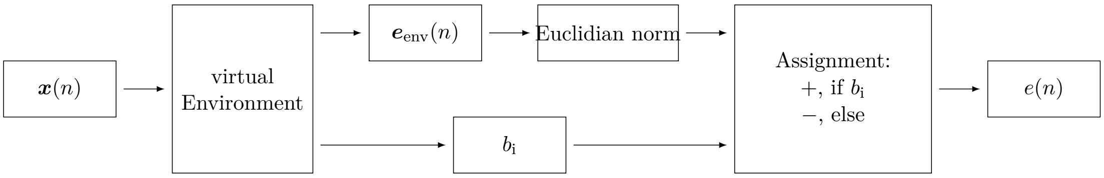


Figure 4.10: The calculation of $e(n) forthemagnitude - basedapproach$

The scalar value for $f_{\text{SFA}}(n)$ yielded by the application of SFA on this scalar $e(n)$ will get multiplied with the unit vector of $\boldsymbol{e}_{\text{initial}}(n)$. In this step, it will also regain all the previously disregarded information on direction.

4.5.1 Determining when the haptic interaction point has left a virtual object

Again, the question arises on how to determine if a position outside the original virtual object is still inside its artificially increased boundaries.

In this case, the magnitude of the vector from the HIP to the CSP can directly be compared to the offset from SFA. As long as the virtual offset is greater, the HIP is still inside the virtually shifted boundaries of the object and there are still forces to be displayed. Only after the distance to the original boundaries exceeds the offset will the interaction be completed. Fortunately, all of this already happens intrinsically in the SFA implementation, so it is sufficient to simply feed it the pre-processed $e(n)$.

4.6 Adding virtual friction

It could be very constructive to display friction on a virtual object's surface. This could help widely in providing more realistic haptic feedback. Especially in combination with an implementation of the computation of the penetration depth proposed in § 4.3.3, it should yield very impressive results.

For the implementation of virtual friction based on the Amonton-Coulomb model, the first step is to find the portion of the HIP's movement which is perpendicular to the virtual object's surface. Finding the direction that is perpendicular to the surface can be very challenging itself, see § 4.3. Assuming this can be done, the next step is to determine the maximum tangential force possibly resulting from friction. This can be done very easily by just declaring a friction coefficient for the entire virtual object, or alternatively a individual value for every displayed virtual object, maybe even for every single surface. Then, the magnitude of the maximum friction force is simply the product of the normal force with that friction coefficient.

In case the tangential partition force exerted by the user does not exceed the maximum friction force, the exact same force should be displayed at the HIP, but in the opposite direction. As soon as the user's tangential force reaches a value exceeding the computed maximum friction force, the resulting displayed force is limited to this maximum.

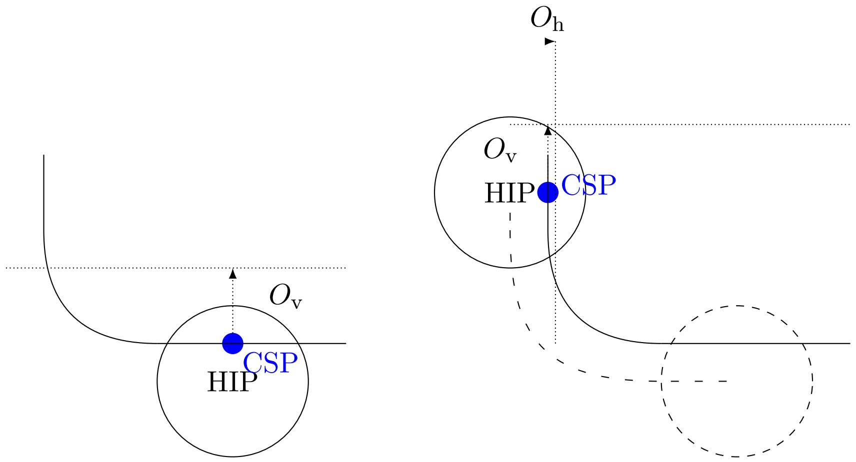
This could be realized similarly to the normal forces of the interaction. Here, it is especially important to achieve a stable interaction because the friction applies to all movement tangential to the virtual object's surface. This means that the HIP might oscillate significantly as a result of opposing friction forces. To further increase the realism, a separate value for the friction coefficient can be chosen for both dynamic and static contact. It is important to set the kinetic friction coefficient to a lower value than the static one. This is done to avoid discontinuities where a higher kinetic friction force stops the tangential movement, so the lower static friction coefficient is used to calculate further, lower friction forces that allow the HIP to move again, which gets immediately stopped again by the higher friction. This mechanism would result in noticeable oscillations.

4.7 Theoretical comparison between magnitude- and concatenation-based extension of the successive force augmentation approach to multiple degrees of freedom

Both approaches are inherently reasonable. The major difference is the fact that the vectorial concatenation results in decoupling especially in the memory attributes such as the virtual offset. The main benefit of the decoupled way to compute the kinaesthetic feedback forces is the opportunity to set different values for the goal stiffness for the individual dimensions. This might be useful in some cases, but is entirely impossible using the magnitude-based approach, as it requires different offsets to display different stiffnesses.

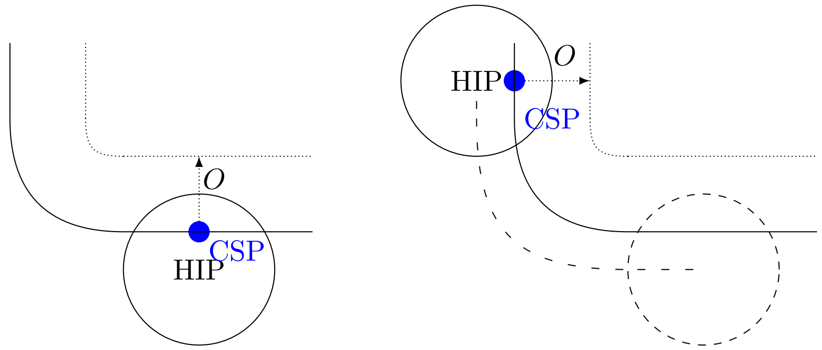
There are also benefits in having an offset independent of direction. For instance, when approaching a virtual object, after the initial contact, which displays the desired stiffness precisely, the displayed stiffness successively rises to the desired value again. To this point, both approaches would yield very similar results.

However, if the user now decides to move along the virtual object's three-dimensional surface, differences in the implementations might gain relevance. Assuming the force directed into the object is constant, the magnitude-based extension would continuously provide the desired stiffness after the first adaption. For the vectorial concatenation however, the forces would shift to a different dimension, which then requires an additional adaptation period where the offset in the previously loaded dimension would get successively reduced while in the currently loaded dimension, the offset would again need to accumulate.



(a) The vertical offset O_v has adapted to display a high vertical stiffness.

(b) Now a high horizontal force is required. The vertical offset can not change anymore and the horizontal one (O_h) first needs to accumulate.



(c) The offset has adapted to display a high vertical stiffness.

(d) Because the offset is not coupled to a certain direction, the desired horizontal stiffness does not require additional adaptation.

Figure 4.11: Different types of offset memory: a and b display a concatenation-, c and d a magnitude-based approach

The same effect also results in friction-like phenomena. The calculated force in any way depends on the virtual offset. Of course, the virtual offset is less relevant for lower stiffness and smaller penetration depth, as the offset required to display the desired stiffness is comparatively small then, because the virtual stiffness itself already generates close to sufficient forces. For high stiffnesses however, the majority of the displayed force results from the offset. As long as this offset has not adapted to a different orientation in the concatenation-based implementation, the displayed force would be directed in a direction that is not parallel to $\boldsymbol{e}(n)$ and therefore not perpendicular to the virtual object's surface. This means there is a nonzero portion of the displayed force that is tangential to the surface and could thereby be perceived as friction or even the virtual object moving. In any way, it distorts the user experience and reduces transparency. The magnitude-based approach will always yield a force that is exactly parallel to $\boldsymbol{e}(n)$, so this phenomenon can not be experienced with this approach.

Chapter 5

Experimental results

In order to verify the results from this work and to gain further insights into the actual feeling the presented approaches provide, the controllers have been implemented in two inherently different haptic devices. Thereby, their robustness regarding running on entirely contrasting mechanical systems could be effectively evaluated.

Although the used systems are different in many ways, one common factor for these experiments was the control frequency, which was set to $1000Hz$.

5.1 Experimental comparison of the enhanced successive force augmentation approach with its original form, the time-domain passivity approach and the force bounding approach in a single degreee of freedom

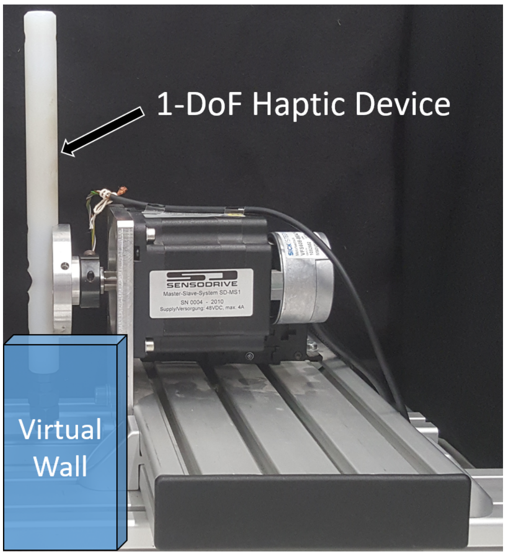


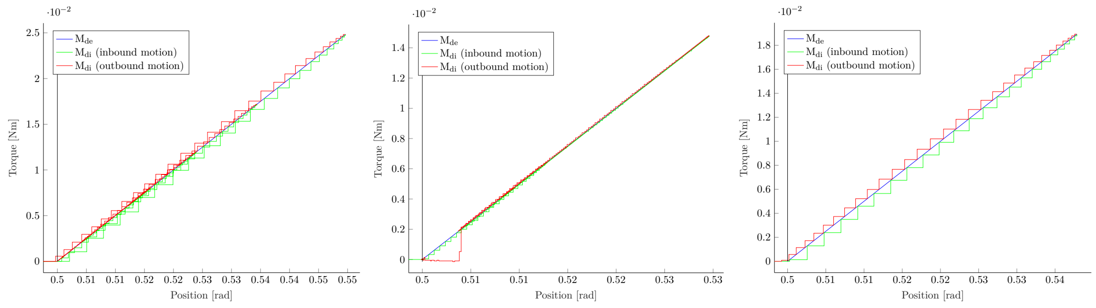
Figure 5.1: The haptic interface that has been used for the experiments in one degree of freedom. The displayed virtual wall was positioned at $x_\mathrm{s} = 0.5rad$.

For the validation of the different implementations even before extending them to 3 DoF, a device with just a single DoF in rotation has been used, which is shown in Fig. 5.1. Using the previously proposed Algorithm 2.1, the inherent physical damping has been found to be $c_\text{i} = 0.0005\frac{Nm\cdot s}{rad}$. The resulting critical stiffness is exactly $k_\text{c} = 1\frac{Nm}{rad}$.

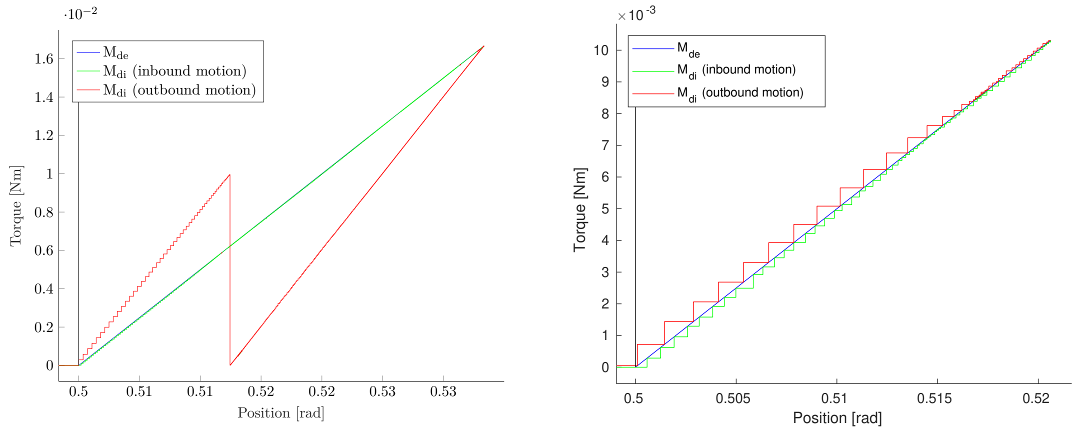
Four different values for the stiffness of the virtual object have been evaluated with each of the approaches: $k_\text{de} \in \{0.5; 1; 2.5; 5\} \cdot \frac{Nm}{rad}$. These stiffnesses have been chosen to include one value lower than k_c, the critical stiffness itself, and two higher values where the uncontrolled virtual spring shows instabilities.

5.1.1 Virtual environment with low stiffness

It is very useful to use the uncontrolled approach as a baseline to evaluate the more complex approaches. At this low stiffness ($k_\text{de} = 0.5\frac{Nm}{rad}$), the energy generated through discretization errors is completely dissipated by the internal damping, so even the uncontrolled spring is stable. For the same reason, the computed force is within the boundaries set by FBA, so its plot is very similar.



(a) virtual spring (b) TDPA (c) FBA



(d) original SFA (e) enhanced SFA

Figure 5.2: Comparison of different approaches displaying a low stiffness.

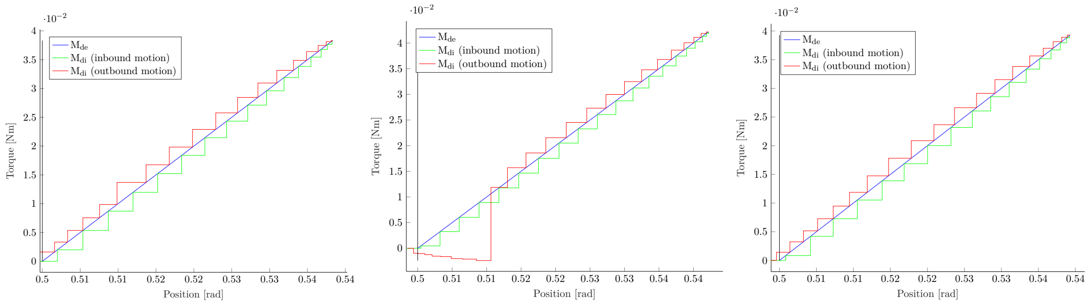
The results of the TDPA implementation show very clearly how the computed energy reaches zero before the HIP leaves the virtual environment because the internal damping is not considered. In consequence, the adaptive damping engages. This even results in negative forces that lead to the perception of an adhesion-like phenomenon.

Interestingly, the original SFA shows a discontinuity in the displayed force as explained in § 3.1. The enhanced version however shows the force exactly following the intended value because it is reduced to an uncontrolled spring for this low stiffness.

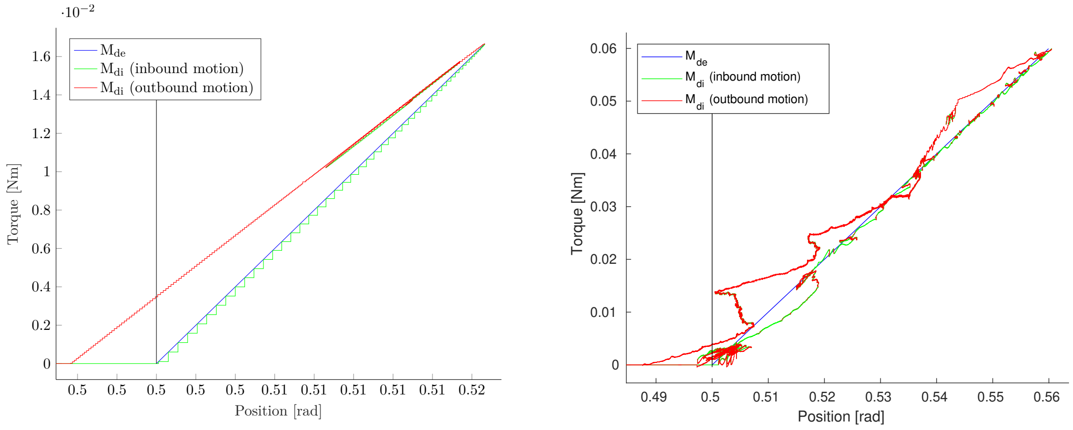
5.1.2 Virtual environment with critical stiffness

The FBA should theoretically be stable for any arbitrarily chosen stiffness. Of course, that only works as long as c_i is chosen correctly.

Differently than before however, the enhanced SFA does not simply display an uncontrolled spring because of its low virtual stiffness $k_\mathrm{v} < k_\mathrm{c} = k_\mathrm{de}$. It can be seen very clearly how the force is continuously adapted to approach the desired value within the limitations posed to ensure a continuous and stable interactions, whereas the original approach clearly generates energy in this interaction.



(a) virtual spring (b) TDPA (c) FBA



(d) original SFA (e) enhanced SFA

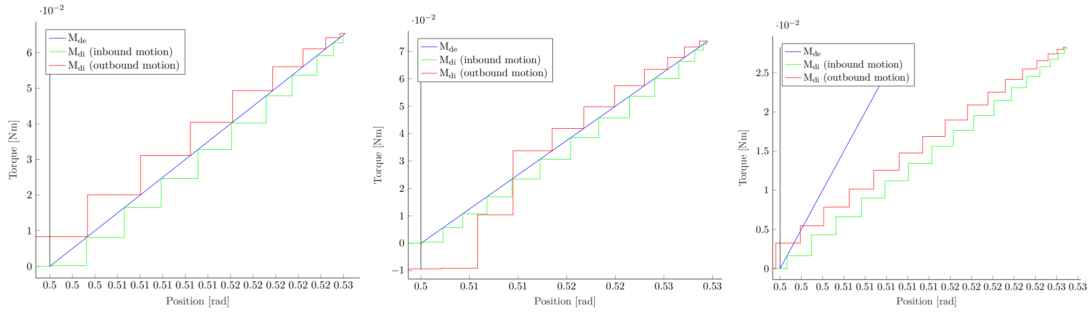
Figure 5.3: Comparison of different approaches displaying the critical stiffness.

5.1.3 Virtual environment with high stiffness

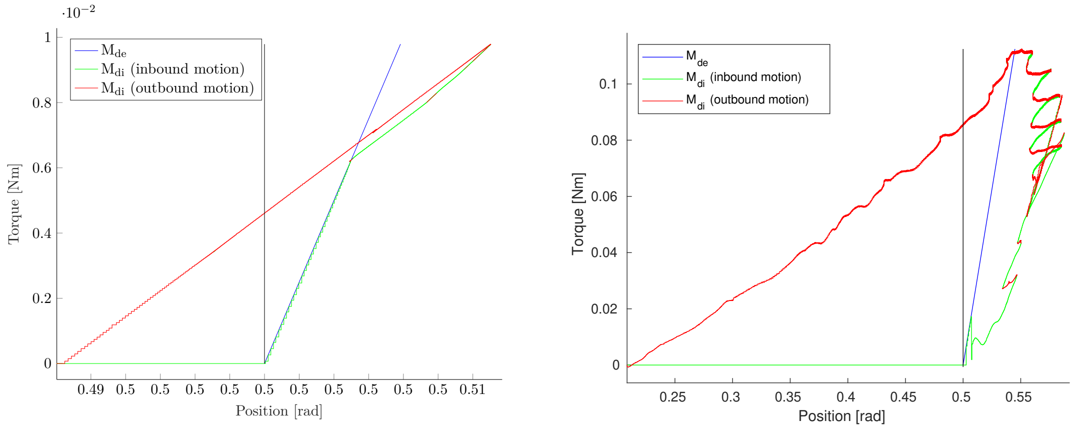
The plot for the uncontrolled virtual spring (Fig. 5.4) shows clearly that a higher stiffness results in higher energy generation caused by discretization error by displaying very large steps that cover a large area between them. This illustrates why generally, a stiffer virtual environment leads to more instability due to the device being active.

The original SFA shows significant energy generation again. Another interesting point to be seen here is that the enhanced approach advances significantly more quickly to the desired torque than the original one. While some difference in the offset is notable between the inbound and the outbound motion for the original approach, it would take very long to adapt the offset to a given position and stiffness, whereas the enhanced approach can be observed to successively reach it with little motion in Fig. 5.4. Fig. 5.5 shows how the HIP's equilibrium point for the applied torque (around $0.1Nm$) is pulled out progressively until the desired stiffness is displayed.

Another very interesting observation is that the enhanced approach comes with the drawback of a considerably larger artificial offset. Of course, this is partially caused by the fact that the original approach takes longer to increase the offset. More importantly however, the enhanced SFA contains the serially interconnected filter. As a result, the displayed force leads to the elongation of two virtual springs instead of just one.



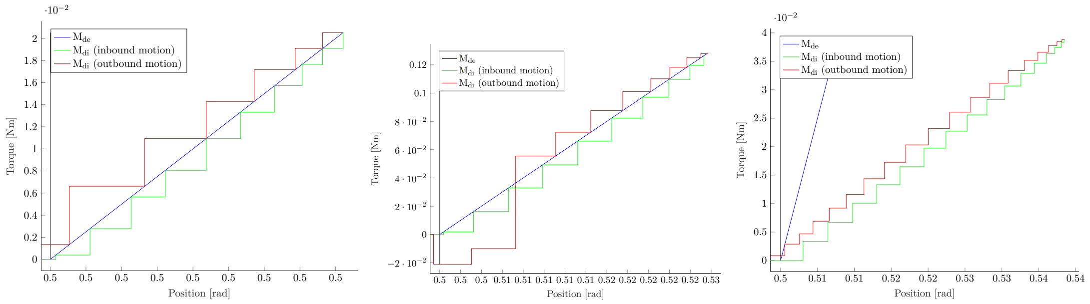
(a) virtual spring (b) TDPA (c) FBA



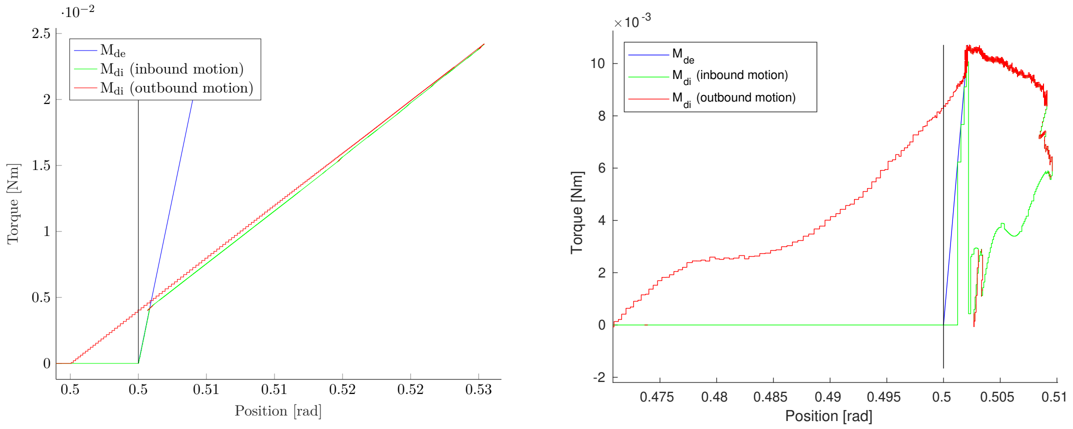
(d) original SFA (e) enhanced SFA

Figure 5.4: Comparison of different approaches displaying a high stiffness $k_{\text{de}} = 2.5\frac{Nm}{rad}$.

While Fig. 5.5 shows only a single interaction for each of the tested approaches, Fig. 5.6 reveals how the uncontrolled virtual spring, the original SFA approach and the TDPA all display significant oscillations. The reason for the virtual spring being unstable at high stiffness is energy generation through discretization error as described in § 1.2.6. For the original SFA approach, it has been explained in § 3.2 that the lower stiffness used on the outbound motion leads to energy generation. The oscillations in TDPA have been described in § 2.1.



(a) virtual spring (b) TDPA (c) FBA



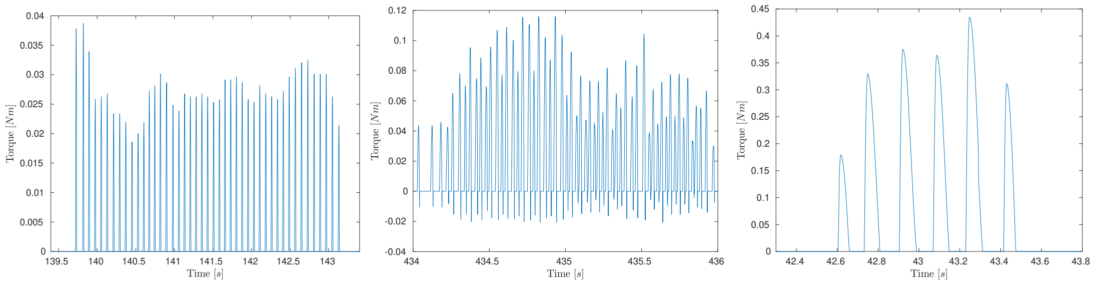
(d) original SFA (e) enhanced SFA

Figure 5.5: Comparison of different approaches displaying a very high stiffness $k_{\text{de}} = 5\frac{Nm}{rad}$.

The figures for FBA show precisely how the actually displayed stiffness is limited to k_{c}, so there are virtually no differences in the interactions at $k_{\text{de}} = 5\frac{Nm}{rad}$, $k_{\text{de}} = 2.5\frac{Nm}{rad}$ and $k_{\text{de}} = 1\frac{Nm}{rad}$. This, of course, ensures stability. However, it effectively restricts the displayable stiffness bandwidth to $k_{\text{de}} \leq k_{\text{c}}$. Therefore, this approach is not suitable to display inherently unstable stiffnesses transparently.

As shown in Fig. 5.7, the rise of the displayed stiffness for the enhanced SFA approach slows down before considerably accelerating again. The reason for this phenomenon is the computed energy reaching zero and the passivity controller enabling the adaptive damping. This results in oscillations as described in § 3.5. Although these oscillations are not perceivable to the user because of the filter, they still cause very frequent updates of the virtual offset, so it rises quickly even if the HIPseems to be static or even moves out of the virtual object.

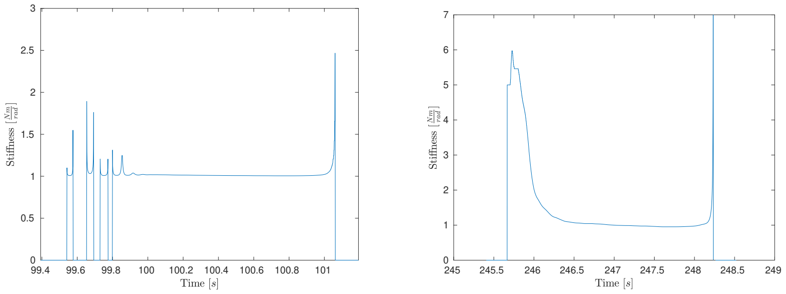
An obvious similarity of the original and the enhanced SFA is the correctly displayed rate-hardness for the initial contact. Both approaches show how the displayed stiffness initially reaches the desired stiffness instantly. After the first motion into the virtual object, the approaches start to considerably differ. The same interaction displayed in Fig. 5.5 is also shown in Fig. 5.7. The original SFA switches to k_{v} immediately after the initial contact. Further movement into the virtual object results in a remarkable decrease of the displayed stiffness. The force augmentation is hardly perceivable. The enhanced implementation in contrast adapts the displayed force very quickly to the current HIP



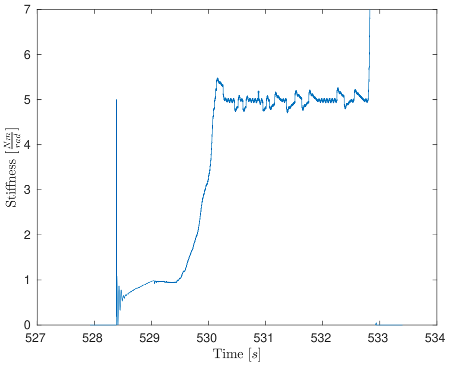
(a) virtual spring (b) TDPA (c) original SFA

Figure 5.6: Different approaches unstably displaying a very high stiffness $k_{\text{de}} = 5\frac{Nm}{rad}$.

position, so the desired stiffness is reached. On the outbound motion, both versions of SFA run through the singularity at $e = 0$ where the displayed stiffness is undefined and changes its sign.



(a) FBA (b) original SFA



(c) enhanced SFA

Figure 5.7: Different approaches stably displaying a very high stiffness. Note however that the FBA doesn't really manage to display the desired stiffness $k_{\text{de}} = 5\frac{Nm}{rad}$, and the original SFA approach is only stable because of considerable manual damping of the operator. This is done to provide a comparison on the response time required to reach the desired stiffness with the enhanced approach.

5.1.4 Conclusion of the one-dimensional experiments

For a very low stiffness $k_{\text{de}} = 0.5\frac{Nm}{rad}$, all of the examined approaches provided stable interactions and managed to display the desired stiffness. The original SFA and TDPA both displayed a discontinuity in the force.

At the critical stiffness $k_{\text{de}} = 1\frac{Nm}{rad}$, all approaches successfully provided the desired stiffness. The original SFA could be observed to be active, but still stable. The enhanced SFA showed some alterations to the desired stiffness because it needs to continuously update the artificial offset in order to reach the proper force for the current position.

For the higher stiffnesses $k_{\text{de}} = 2.5\frac{Nm}{rad}$ and $k_{\text{de}} = 5\frac{Nm}{rad}$, the uncontrolled virtual spring, the original SFA and the TDPA behaved unstably, although all of them still mostly displayed the desired stiffness. The FBA was stable, but only displayed the critical stiffness $k_{\text{c}} = 1\frac{Nm}{rad}$. The enhanced SFA provided a stable interaction, displayed the desired stiffness accurately on the first interaction to ensure the correct rate-hardness, and finally progressively approached and reached the desired stiffness asymptotically.

Overall, the enhanced SFA could be proven to be an actual improvement over the original approach. In these experiments, it successfully displayed different stiffnesses in a stable way.

5.2 Experimental comparison of the proposed three-dimensional implementations of the successive force augmentation approach

Figure 5.8: The haptic interface that has been used for the experiments in three degrees of freedom. Image from Force Dimension [8].

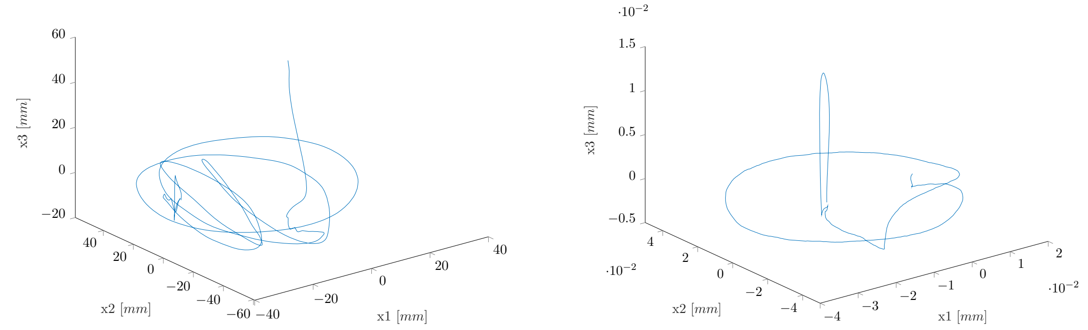
The device used to conduct the three-dimensional experiments was a ForceDimension omega.3, as pictured in Fig. 5.8. This device provides 3 DoF that are entirely translational due to its delta robot structure. Another important benefit of this structure is the low inertia resulting from all the acutators being positioned in the robot base. Using FBA in a single degree of freedom, the inherent physical damping of this device could be identified to a value of $c_\mathrm{i} = 0.0025\frac{N \cdot s}{mm}$. The corresponding critical stiffness equates to $k_\mathrm{c} = 5\frac{N}{mm}$.

Three different environments have been chosen to validate the achieved results in multiple steps. These environments differ considerably in complexity. While some will be more demanding and thereby uncompromisingly show potential flaws in an implementation, others will be less complex but provide more detailed insights into the variables involved in the approach.

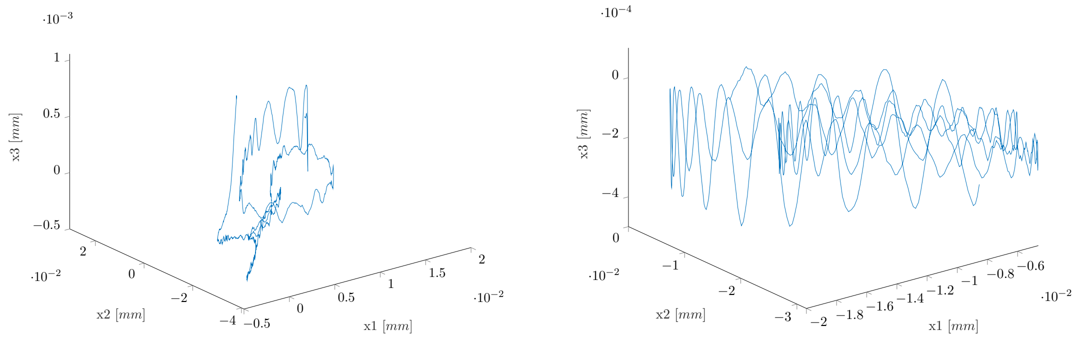
5.2.1 Horizontal plane

The least complex and thereby easiest environment to analyze simply consisted of a virtual plane through the $x1$- and the $x2$-axis of the haptic interface. This virtual environment isolates just a single dimension for forces to be generated in. In consequence, the approach's general validity can be analyzed very easily because only a single dimension needs to be considered.

This environment has been used to show the behavior for an uncontrolled virtual spring as a reference, too. The results shown in Fig. 5.9 illustrate that for stiffnesses higher than the critical stiffness ($k_\mathrm{c} = 5\frac{N}{mm}$), the interaction can become unstable very quickly.



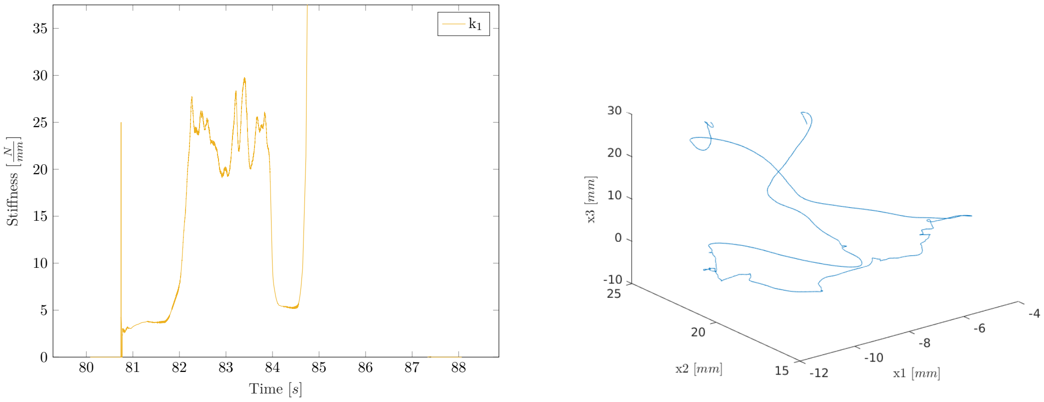
(a) $k_\mathrm{de} = 2.5\frac{N}{mm}$ (b) $k_\mathrm{de} = 5\frac{N}{mm}$



(c) $k_\mathrm{de} = 10\frac{N}{mm}$ (d) $k_\mathrm{de} = 25\frac{N}{mm}$

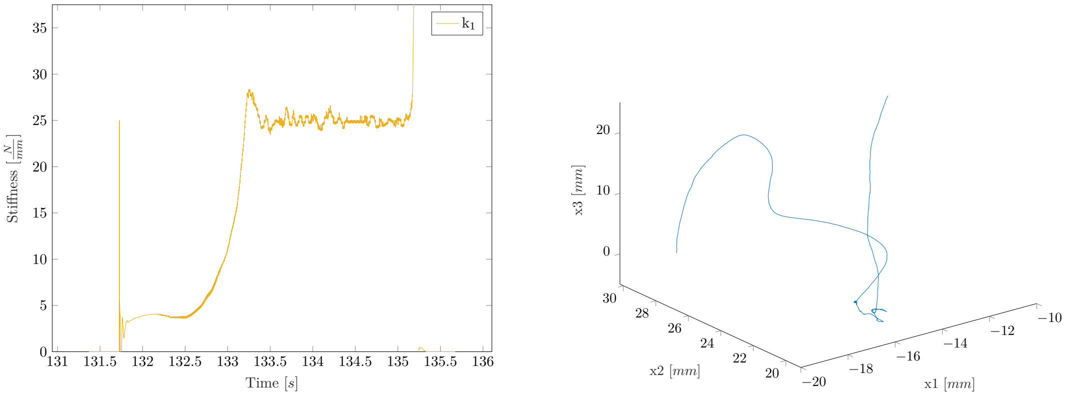
Figure 5.9: Comparison of different stiffnesses with an uncontrolled virtual spring.

Both SFA approaches yield very similar results as shown in Fig. 5.10. This was already expected, as they are mathematically equivalent for this case. The interaction is completely stable for both implementations. On the initial contact, the rate-hardness is properly displayed.



(a) concatenated approach, $k_{\text{de}} = 25\frac{N}{mm}$, displayed stiffness

(b) concatenated approach, $k_{\text{de}} = 25\frac{N}{mm}$, HIP trajectory



(c) magnitude-based approach, $k_{\text{de}} = 25\frac{N}{mm}$, displayed stiffness

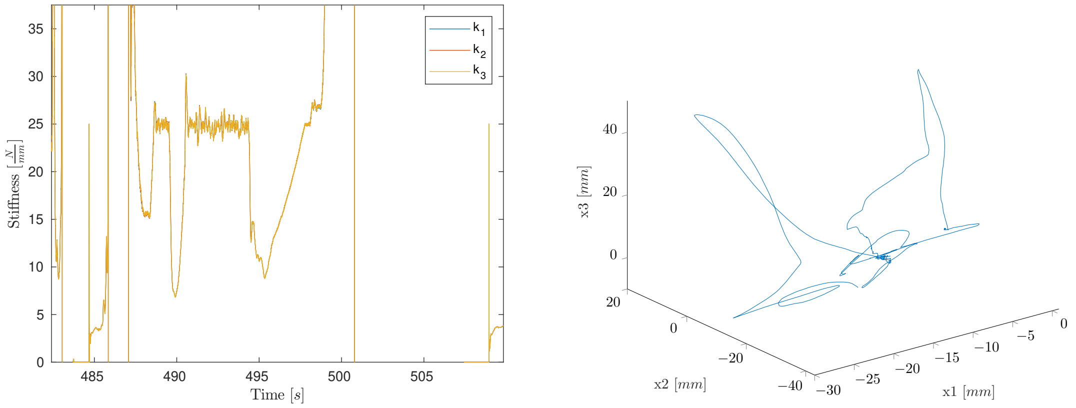
(d) magnitude-based approach, $k_{\text{de}} = 25\frac{N}{mm}$, HIP trajectory

Figure 5.10: Comparison of the concatenated approach with the magnitude-based one on a virtual plane. The individual values for the stiffness are derived from the division of the displayed force with the penetration depth for each coordinate.

Unfortunately, this virtual environment does not inspect the implementation's behavior in response to combinations of the three translational dimensions. Therefore, it is not sufficient to completely validate the approach.

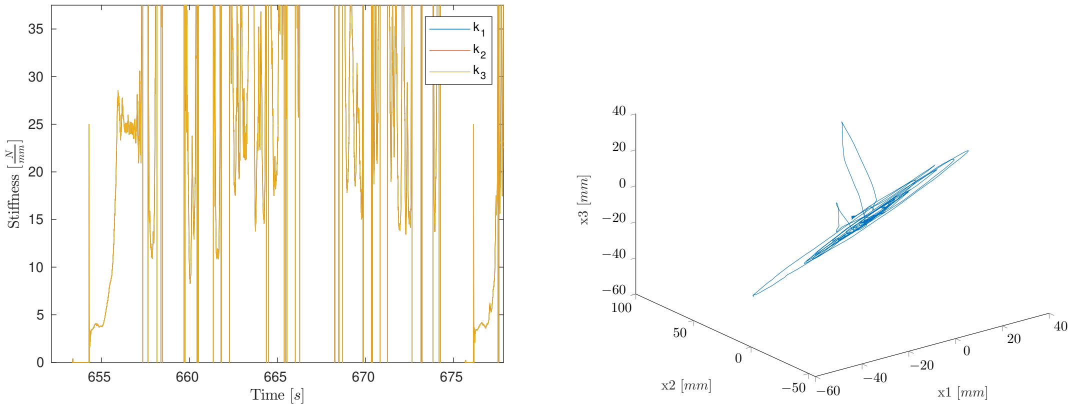
5.2.2 Inclined plane

A slightly more complex virtual environment has been achieved by creating an inclined plane that does not include any of the coordinate frame's axes. This environment provides forces in all three translational dimensions simultaneously and in different magnitudes. Therefore, the implementation's response to a multidimensional $\boldsymbol{e}[n]$ can be closely observed here. At the same time, it is still a plane, so the force's (desired) direction is constant. Thus, this environment is very suitable to examine the implementations generation of forces primarily regarding their magnitude.



(a) concatenated approach, $k_{\text{de}} = 25\frac{N}{mm}$, displayed stiffness

(b) concatenated approach, $k_{\text{de}} = 25\frac{N}{mm}$, HIP trajectory



(c) magnitude-based approach, $k_{\text{de}} = 25\frac{N}{mm}$, displayed stiffness

(d) magnitude-based approach, $k_{\text{de}} = 25\frac{N}{mm}$, HIP trajectory

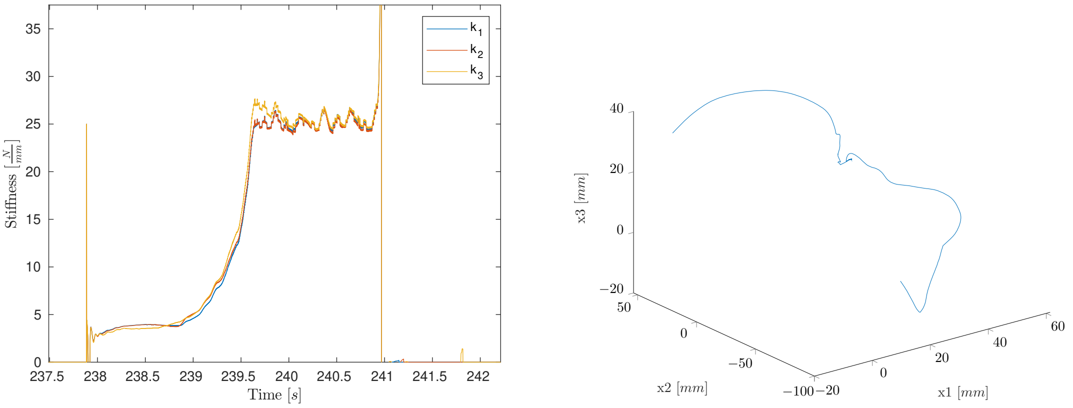
Figure 5.11: Comparison of the concatenated approach with the magnitude-based one on an inclined virtual plane.

The results from this environment show clearly how the direction of the forces is irrelevant to both the implementations as long as its constant. Only to changes in this direction will the different approaches respond differently.

It is interesting to see how the displayed stiffness is affected by lateral movement of the HIP because the normal forces are unavoidably altered during this process.

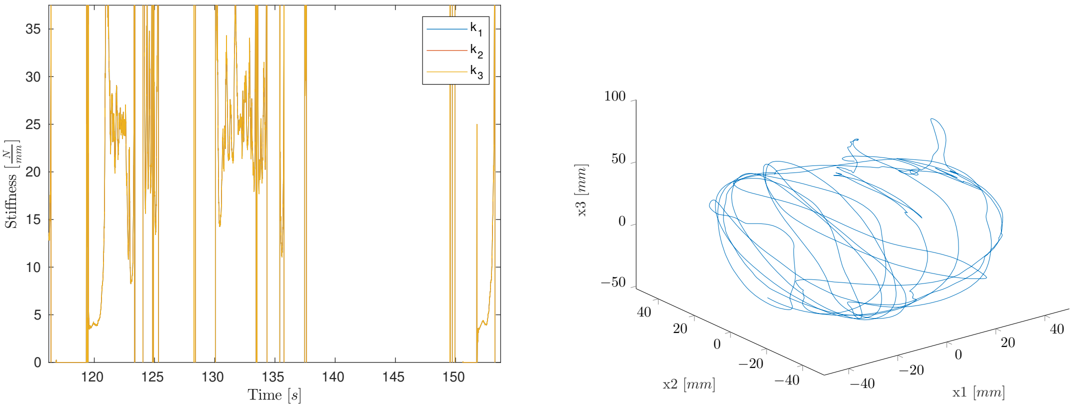
5.2.3 Sphere

The final virtual environment used in the experiments contained a virtual sphere placed in the center of the haptic interface's workspace. Although this object is comparatively simple, it still provides very valuable insights to the approaches' response to variable directions of $\boldsymbol{e}[n]$. All of the three dimensions of translation as well as every possible linear combination of them are covered by this test, as the sphere's surface continuously incorporates all of them. The major drawback of this experiment is that the results are inherently multidimensional and can not be easily simplified for analysis.



(a) concatenated approach, $k_{\text{de}} = 25\frac{N}{mm}$, displayed stiffness

(b) concatenated approach, $k_{\text{de}} = 25\frac{N}{mm}$, HIP trajectory



(c) magnitude-based approach, $k_{\text{de}} = 25\frac{N}{mm}$, displayed stiffness

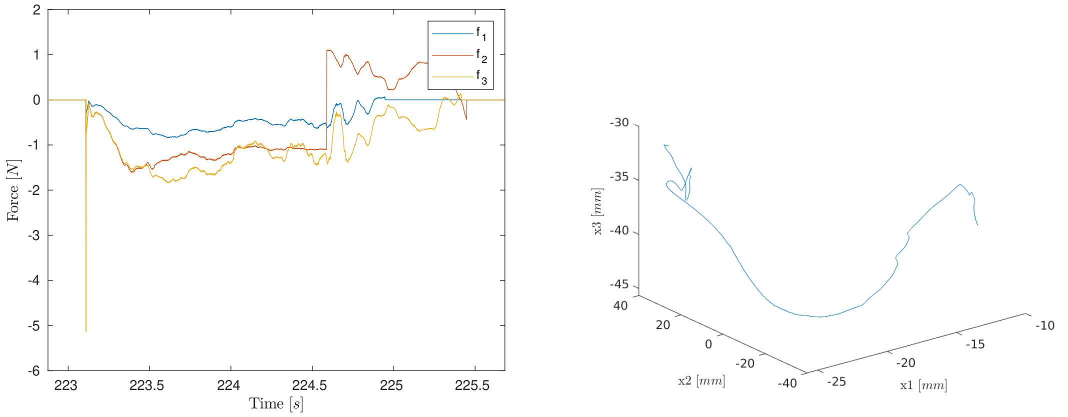
(d) magnitude-based approach, $k_{\text{de}} = 25\frac{N}{mm}$, HIP trajectory

Figure 5.12: Comparison of the concatenated approach with the magnitude-based one on a virtual sphere.

This environment shows the differences between the different approaches very clearly. While both implementations successfully reached the desired stiffness on perpendicular probing of the virtual sphere as shown in Fig. 5.12, they produced very differing results for tangential movement while in contact with the object.

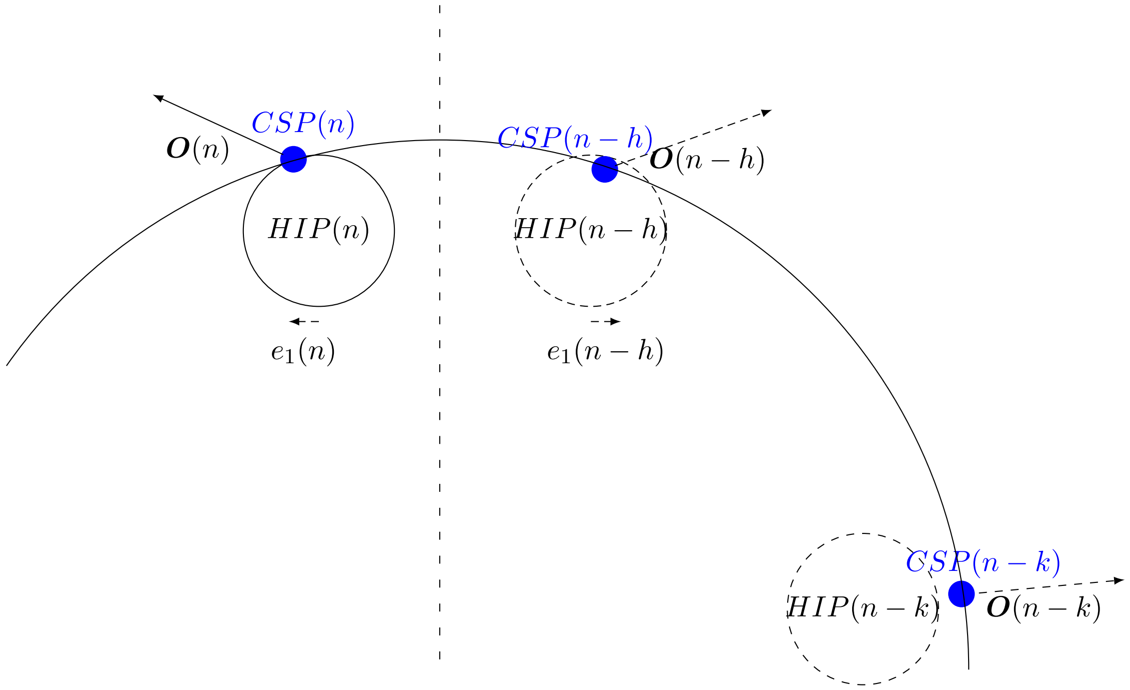
In particular, the decoupled values for the artificial offset in the concatenated implementation considerably distorted the interaction. Fig. 5.13 shows a very distinct problem that is quite similar to what was described in Fig. 4.11. This is the reason for the stiffnesses being different for different coordinates as seen in Fig. 5.12.

In contrast to those difficulties for the concatenated approach, the magnitude-based one can be observed to quickly reach the desired stiffness not only on the initial contact, but also after and even while some movement takes place (Fig. 5.12).



(a) concatenated approach, $k_{\text{de}} = 25\frac{N}{mm}$, displayed forces. It is clearly visible how the displayed force in the $x2$-dimension suddenly changes its direction when it should actually be zero.

(b) concatenated approach, $k_{\text{de}} = 25\frac{N}{mm}$, HIP trajectory



(c) At $n-k$, the offset has adapted to the current position. After some movement $(n-h,\, h<k)$, the offset is not adequate anymore. Instead, an improper force that is not perpendicular to the expected virtual surface can be perceived. After some more movement (n), the direction of e_1 has changed, which O_1 is based on so O it will always point outside the virtual environment. As a result, the force in this dimension also jumps between $n-h$ and n from a positive to a negative value abruptly. In the ideal case, the force in this dimension would be exactly zero at this point, so the direction change would not be perceivable.

Figure 5.13: A major drawback of the decoupled memory

5.2.4 Conclusion of the three-dimensional experiments

For simple planes, both implementations achieved very similar results that were stable and quickly reached the desired stiffness after displaying it on initial contact to ensure the desired rate-hardness. Even on the inclined plane, both approaches had promising and similar results, therefore it can be concluded that the angle of the displayed force does not correlate with distorted interaction.

The virtual sphere however showed the drawbacks of the concatenation-based approach distinctly. The magnitude-based approach yielded significantly more transparent interaction and no perceivable discontinuities in the displayed force, whereas the concatenated implementation suffered from its direction-dependent memory.

Chapter 6

Conclusion

6.1 Summary of results

This thesis proposed several enhancements to the SFA approach:

Stabilizing the rate-hardness display by allowing a discontinuous force

Reducing the response time by updating the offset more frequently by larger steps

Ensuring passivity by implementing the TDPA

Removing oscillations resulting from the TDPA with a virtual filter

Subsequently, the SFA approach has been extended to three translational DoF. An application of the control approach on the magnitude of the feedback force vector and a separate application of the approach to every single translational dimension have been examined here. A theoretical comparison suggested that the magnitude-oriented implementation might provide more transparent force feedback.

Experiments have been conducted to validate the enhanced approach and compare it to the original SFA approach, TDPA, FBA and a virtual spring without any additional controllers in a rotational single-DoF haptic interface. They suggest significant improvements over the original approach.

A three-dimensional haptic interface covering all three dimensions of translation has been used to validate the multidimensional extensions of the SFA approach. The experimental results collected on the omega.3 device support the previous theoretical analysis of the different implementations: It could be concluded that the application of SFA on the force vector's magnitude yields better results than the decoupled implementation.

Furthermore, the experiments on those two devices differing in DoF, system dynamics, sensor resolution, workspace as well as maximum displayable force and torque respectively confirm the robustness of the developed control approach to significantly varying conditions.

6.2 Future work

Different approaches on how to find a reasonable penetration vector $\boldsymbol{e}$ to base the computation of adequate feedback forces on have been explained. However, all the different environments used in the experiments have in common that the way $\boldsymbol{e}$ gets determined is very simple and only the CSP has been used in all of them. This means that although those different approaches have been compared theoretically, they have not been validated through actual experiments.

Apart from that, there might be even more elaborate and different methods to determine this value that have not been mentioned here.

As described in § 4.6, the implementation of friction could enhance the user experience. This has not been implemented or tested here, but could be a major improvement for kinaesthetic feedback in virtual reality.

In § 4, this thesis described approaches to implement the concept of SFA in multi-dimensional applications. However, this was limited to only three dimensions consisting entirely of translation. Of course, it would be interesting to extend the implementation to all six dimensions of the task space.

Finally, the SFA approach might be suitable to be extended for telerobotics.

Appendix A

Notation

A.1 Abbreviations and acronyms

CSP closest surface point

DoF degrees of freedom

FBA force bounding approach

HIP haptic interaction point

POP point of penetration

SFA successive force augmentation

TDPA time-domain passivity approach

List of Algorithms

List of Figures

References

[1] Karl Arthur, Andrew J Doxon, Courtney Parsons, and William R Provancher. Lock picking simulation using visual and bimanual haptic display. In *2012 IEEE Haptics Symposium (HAPTICS)*, pages 547–551. IEEE, 2012.

[2] Arvind Balijepalli and Thenkurussi Kesavadas. A haptic based virtual grinding tool. In *11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings.*, pages 390–396. IEEE, 2003.

[3] Tales Nereu Bogoni, Gregory de Oliveira Feijó, Roberta Kochenborger Scarparo, and Márcio Sarroglia Pinho. Endodontic simulator for training the access to the pulp chamber and root canal preparation tasks. pages 523–525. IEEE, 2016. ISBN 978-1-4673-9042-2. doi: 10.1109/ICALT.2016.51.

[4] J.E. Colgate, M.C. Stanley, and J.M. Brown. Issues in the haptic display of tool use. volume 3, pages 140–145 vol.3. IEEE, 1995. ISBN 0-8186-7108-4. doi: 10.1109/IROS.1995.525875.

[5] Dana D. Damian, Marvin Ludersdorfer, Yeongmi Kim, Alejandro Hernandez Arieta, Rolf Pfeifer, and Allison M. Okamura. Wearable haptic device for cutaneous force and slip speed display. pages 1038–1043. IEEE, 2012. ISBN 978-1-4673-1404-6. doi: 10.1109/ICRA.2012.6225000.

[6] ESA and DLR. Analog-1. accessed March 11, 2020, November 2019. URL `https://www.esa.int/ESA_Multimedia/Images/2017/11/Analog-1`.

[7] David Escobar-Castillejos, Julieta Noguez, Luis Neri, Alejandra Magana, and Bedrich Benes. A review of simulators with haptic devices for medical training. *Journal of medical systems*, 40(4):104, 2016.

[8] Resources from Force Dimension. omega.3. URL `https://www.forcedimension.com/images/presskit/omega.3-high.jpg`. accessed March 11, 2020.

[9] Patrick Gifford. The forgotten johann philipp reis, June 2015. URL `https://www.inc-installs.com/technology/the-forgotten-johann-philipp-reis`. accessed March 11, 2020.

[10] Yoav Golan, Ben Serota, Amir Shapiro, Oren Shriki, and Ilana Nisky. A vibrotactile vest for remote human-dog communication. pages 556–561. IEEE, 2019. ISBN 978-1-5386-9462-6. doi: 10.1109/WHC.2019.8816079.

[11] Jong-Phil Kim, Sang-Yun Baek, and Jeha Ryu. A force bounding approach for multi-degree-of-freedom haptic interaction. 20:1193–1203, 2015. ISSN 1941-014X. doi: 10.1109/TMECH.2014.2333537.

[12] Myungsin Kim, Inyoung Jang, Yongseok Lee, Yongjun Lee, and Dongjun Lee. Wearable 3-dof cutaneous haptic device with integrated imu-based finger tracking. pages 649–649. IEEE, 2016. ISBN 978-1-5090-0822-3. doi: 10.1109/URAI.2016.7733974.

[13] Ki-Uk Kyung, Seung-Chan Kim, and Dong-Soo Kwon. Texture display mouse: Vibrotactile pattern and roughness display. 12:356–360, 2007. ISSN 1941-014X. doi: 10.1109/TMECH.2007.897283.

[14] D.A. Lawrence, L.Y. Pao, A.M. Dougherty, M.A. Salada, and Y. Pavlou. Rate-hardness: a new performance metric for haptic interfaces. 16:357–371, 2000. ISSN 2374-958X. doi: 10.1109/70.864228.

[15] ByeongKyu Lim, Keehoon Kim, and Donghyun Hwang. On the design of the 5-dof finger-wearable cutaneous haptic device. pages 872–878. IEEE, 2017. ISBN 978-1-5386-3743-2. doi: 10.1109/ROBIO.2017.8324527.

[16] Zih-Hao Lin and Shana Smith. A natural user interface for realistic tactile perception of object surface texture. volume 2, pages 370–373. IEEE, 2017. ISBN 978-1-5386-3023-5. doi: 10.1109/IHMSC.2017.194.

[17] André Martin. Cathode ray tubes for industrial and military applications. *Advances in electronics and electron physics*, 67:198, 1986.

[18] Hermann Mayer, Istvan Nagy, Alois Knoll, Eva U Braun, Robert Bauernschmitt, and Rüdiger Lange. Haptic feedback in a telepresence system for endoscopic heart surgery. 16:459–470, 2007. ISSN 1054-7460. doi: 10.1162/pres.16.5.459.

[19] Gabriel Moy, C Wagner, and Ronald S Fearing. A compliant tactile display for teletaction. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 4, pages 3409–3415. IEEE, 2000.

[20] Won-Hyeong Park and Sang-Youn Kim. Film-type bendable vibrotactile actuator for mobile games. pages 184–185. IEEE, 2017. ISBN 978-1-5090-5545-6. doi: 10.1109/ICCE.2017.7889279.

[21] Harsimran Singh, Aghil Jafari, and Jee-Hwan Ryu. Increasing the rate-hardness of haptic interaction: Successive force augmentation approach. pages 653–658. IEEE, 2017. ISBN 978-1-5090-1426-2. doi: 10.1109/WHC.2017.7989978.

[22] Harsimran Singh, Bhivraj Suthar, Syed Zain Mehdi, and Jee-Hwan Ryu. Ferro-fluid based portable fingertip haptic display and its preliminary experimental evaluation. pages 14–19. IEEE, 2018. ISBN 978-1-5386-5425-5. doi: 10.1109/HAPTICS.2018.8357146.

[23] Harsimran Singh, Dominik Janetzko, Aghil Jafari, Bernhard Weber, Chan-Il Lee, and Jee-Hwan Ryu. Enhancing the rate-hardness of haptic interaction: Successive force augmentation approach. 67:809–819, 2020. ISSN 1557-9948. doi: 10.1109/TIE.2019.2918500.

[24] Femke E. van Beek, Dennis J.F. Heck, Henk Nijmeijer, Wouter M. Bergmann Tiest, and Astrid M.L. Kappers. The effect of damping on the perception of hardness.

pages 82–87. IEEE, 2015. ISBN 978-1-4799-6624-0. doi: 10.1109/WHC.2015.7177695.

[25] Ramiro Velázquez, Edwige E Pissaloux, and Michael Wiertlewski. A compact tactile display for the blind with shape memory alloys. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 3905–3910. IEEE, 2006.

[26] Gi-Hun Yang, Ki-Uk Kyung, Mandayam A Srinivasan, and Dong-Soo Kwon. Quantitative tactile display device with pin-array type tactile feedback and thermal feedback. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 3917–3922. IEEE, 2006.

[27] Tae-Heon Yang, Sang-Youn Kim, and Dong-Soo Kwon. Applications of a miniature pin-array tactile module for a mobile device. In *2008 International Conference on Control, Automation and Systems*, pages 1301–1304. IEEE, 2008.

[28] Tae-Heon Yang, Sang-Youn Kim, Chong Hui Kim, Dong-Soo Kwon, and Wayne J Book. Development of a miniature pin-array tactile module using elastic and electromagnetic force for mobile devices. In *World Haptics 2009-Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 13–17. IEEE, 2009.

[29] Kazuma Yoshino and Hiroyuki Shinoda. Contactless touch interface supporting blind touch interaction by aerial tactile stimulation. pages 347–350. IEEE, 2014. ISBN 978-1-4799-3131-6. doi: 10.1109/HAPTICS.2014.6775479.

[30] Hongdong Zhang, Yuru Zhang, Dangxiao Wang, and Lei Lu. Dentaltouch: A haptic display with high stiffness and low inertia. pages 388–393. IEEE, 2017. ISBN 978-1-5090-1426-2. doi: 10.1109/WHC.2017.7989933.