A new method to render virtual walls for haptic systems: "Tracking wall". Application to needle insertion simulation

Ma de Los Angeles Alamilla Daniel, Richard Moreau, Tanneguy Redarce

A new method to render virtual walls for haptic systems: “Tracking wall". Application to needle insertion simulation

Ma de los Angeles Alamilla D.[1,*], Richard Moreau[1,**], and Tanneguy Redarce[1,***]

[1] *University of Lyon, INSA Lyon, CNRS, Ampère, F-69621, Villeurbanne, France*

Abstract. For medical training, the use of haptic systems is a way to improve skills of novices, due to their capabilities to reproduce the natural gestures or replicate different environments. A simulator based on a haptic interface allows to reproduce the feeling of inserting a needle and render the different stiffness tissues while a needle is inserted. The majority of needle insertion simulators make use of prosthesis or limbs which limits the training procedures. Some simulators offer more realistic behavior using actual control laws based on active walls. However, problems occur with the speed estimation and the damping factor. To overcome this, an algorithm which can be used in a haptic simulator for the intraarticular punction procedure was developed. This algorithm allows rendering different tissues stiffness while a needle penetrates it avoiding speed estimation problems and rumbling phenomena caused by high damping factors.

1 INTRODUCTION

Haptic simulators are the most suitable solutions, as they allow to manipulate virtual environments with force feedback to increase immersion.

To render forces, haptic simulators may use biomechanical models based on finite element approaches, but due to the heavy resources needed, this can be a limit for further implementation and expansion. Also, the inflexibility to simulate different stiffness that are inherent to the anatomy: skin, fat, muscles and bone are ones of the main problems to overcome [1].

The restrictive movement of the needle is another issue to solve, as these simulators fix the insertion points, losing realism during the process [2].

In this paper, we focus on the intraarticular injection which is one of the most used methods to treat shoulder pain. The insertion of the needle is carried on using anatomical landmarks on the injection site [3]. The rheumatologists use these landmarks to guide their needle inside the shoulder (Fig.(1)).

As the needle is inserted into the body, the main difficulty lies in the fact that it is a blind procedure. To train, doctors practice on corpses or manikins. It helps them to improve their skills before practicing on real patients. However, from an ethical point of view, it is desirable to "never make it first time on a patient" as it is stated by the French H.A.S. (Haute Autorité en Santé) [5], which is an authority in charge of healthcare issues in France. Moreover, this kind of training has limitations due to the fact that training is not repeatable and gesture evaluation is mostly not available. This paper explains and shows first results of a new algorithm, to render the exerted forces on the needle tip when it is inserted into any articulation (shoulder, knee, wrist, etc.), which is a classic procedure for rheumatologists. This method avoids the use of finite element models or methods that need resource consuming processor to render the desired forces on the needle tip.

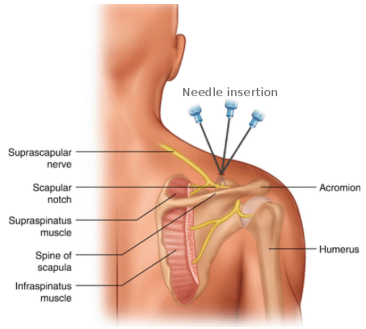

Figure 1. Needle insertion in the shoulder ([4]).

1.1 State of art

Render real haptic feedback of medical procedures is the main objective of many researchers nowadays.

To help doctors to develop skills on how to perform a needle insertion (especially in joints) it is necessary to know what forces are involved and how to render them from the virtual environment to the haptic device.

Considering that the needle shaft rubs against the tissue while the needle cuts the skin, three forces are taken into account (Fig. 2):

*e-mail: ma-de-los-angeles.alamilla-daniel@insa-lyon.fr
**e-mail: richard.moreau@insa-lyon.fr
***e-mail: tanneguy.redarce@insa-lyon.fr

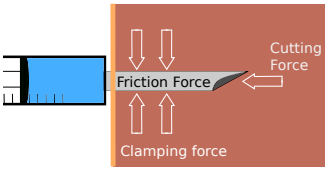

Figure 2. Interaction forces.

1. The cutting force, which acts on the tip of the needle in the axial direction. Its intensity differs according to the various tissue layers

2. The friction force, which acts along the side of the needle shaft in the axial direction.

3. The clamping force, which acts on the side of the needle shaft in the normal direction.

The main force to render via the haptic system is the cutting force when the needle pierces the tissue. This force is affected by the shape of the tip of the needle and the point of insertion as is explained in [6], due to the orientation of the muscle fibers.

One method to render forces in the virtual environment is using the God-Object algorithm, which is implemented in several haptics simulators [7]. This algorithm begins with the assumption that it is not possible to avoid that the user overpasses the virtual location of a surface in the real world. This is due to the inexactitude of the human body to sensor small forces from the haptic device. To avoid the unreal behavior, the algorithm uses a virtual point called the God-Object, which is located at the edge of the virtual object. The forces are rendered using impedance control techniques based on the relative position of the god object against the real tool. This method is simple and useful for solid contact.

The use of advanced software and hardware implementations is commonly found in the bibliography. These methods can be applied to render cutting and friction force, due to both forces are axial. One example of this is the method explained in [8], where the authors used a finite element approach to mesh a 3D model of a virtual liver for a laparoscopic procedure. The method is heavily hardware dependent as it uses GPU units to process the algorithm in parallel. The results show that the haptic training can be performed with a realistic behavior of the laparoscopic procedure, and it is also possible to cut the model in real time as the algorithm introduces new elements into the mesh when this occurs, all of this with a maximum of 70 Fps (Frames per second) of real-time animation. However, as it has been established by the authors, the optimization is a difficult process and the high requirements of hardware limit the method implementation into more medical procedures.

A very common way to render forces for simulating real environments is using deformable surfaces. Using a surface made of elements, the authors of [9] create a deformable mesh that gives the deformation feel feedback to haptic devices. This mesh uses MSM approaches (Mass-spring model) and it is computed using the GPU unit programmed by OpenGL Shading Language to take advantage of the processor calculation power. The results shown by the authors demonstrate that the method can be used in deformable scenarios, such as sphere glide with friction and frictionless environments or deformable sphere fall test. However, this method is very resource consuming, as depend heavily on GPU unit processor and the update is performed each 4 ms.

For underactuated haptic devices, the most used render algorithms are not suited for them. To solve this, a proxy-based haptic rendering is proposed in [10]. In this paper, the authors take a new approach were the haptic forces are rendered using proxy-based method that calculates the forces to render using a deviation in the actuated state from the devices. This leads in no linear equation systems. A linearization process is made to overcome this problem. The method uses the jacobian of the system to calculate the torque applied in the system and can be applied into 3 DoF (degree of freedom) or 6 DoF. However, the passivity and transparency of the system is not ensured, as the authors establish that in some cases, a passivity control much be used as well.

In the case of rendering clamping forces, the use of virtual fixture allows rendering the forces that surround the needle. Virtual fixtures are usefully and very common in the field of haptic simulators, as they offer virtual haptic boundaries that prevent the user going into forbidden areas. Normally, virtual fixtures are set as a soft wall that limits the operation region where the student can perform the task. However, as it has been established by [11], virtual fixtures can also be implemented to set training paths. In their tests, the authors have implemented a path created by 110 points to train a laparoscopic procedure. Although this technique seems to be effective for path tracking, doctors ask for more liberty during training sessions, as each treatment is similar but not equal, as there are a wide variety of factors to be considered.

1.2 Rendering Forces

The fundamental principle of haptic simulators is to identify the position of the virtual element driven by the user, then link it up with the localization of virtual objects that are intended to work, by comparing the coordinates and calculate the possible collisions with virtual objects. The forces and torques are generated to give the sensation of interaction.

With haptic devices, it is possible to build different virtual environments like objects/wall collision scenario. This can be done by controlling the desired force applied by the device on the hand of the user. In theory, this solution can be the most suitable, but in practice, chattering problems will occur on the tip of the tool due to the sudden change of applied force [12].

A virtual spring is a possible solution to render the different body stiffness as long as the stiffness is known [13].

Only one spring can be implemented to simulate the insertion of a needle through one layer. The force deliv-

ered by the spring is proportional to the displacement of the needle, as it is shown in (1).

$$f_R = K(Xo - X) = K\Delta x \tag{1}$$

where f_R represents the cutting force exerted in the virtual needle, Δx represents the thickness of each layer and K the stiffness of the involved layer of the body.

The value of the force f_R can vary in function of the penetrated body part and the distance travelled by the needle [6].

To avoid the chattering problem, walls can be rendered using a virtual stiffness/damping coefficient, modelled in (2).

$$f_R = \begin{cases} K(x(t) - x_{wall}) + B\dot{x} & \text{if } x(t) \geq x_{wall} \\ 0 & \text{if } x(t) < x_{wall} \end{cases} \tag{2}$$

Where K is the stiffness coefficient of the spring, $x(t)$ is the displacement of the haptic tool respect with time, x_{wall} is the position of the wall, B is the viscosity coefficient of the damper, $\dot{x}$ is the velocity of the haptic interface.

As this method is based on Hook's law, the results will demonstrate that as Δx is directly proportional to f_R, and it is necessary that the needle travel a given distance inside the body to get the desired force. Also, as the distance increases inside the body, the force exerted also increases, being at one point, greater than the desired force. This behavior is not realistic since, during needle insertion, the user must feel the desired force and this must be constant until the needle reaches another layer.

Also, this method does not allow to perform stops inside the virtual body. If a stop occurs during the procedure, the spring will keep exerting force, trying to move the virtual needle out of the body if the user does not apply force anymore. To solve these problems, a different method has been developed to create a virtual wall and render the different layers of the body.

2 Tracking wall

Tracking wall approach solves some of the aforementioned problems by implementing a mobile spring (Fig3). Contrary to the last method, where the wall of the spring is fixed, this method allows to exert the same desired force for almost all the needle trajectory, and if a stop occurs the depth of the insertion is memorized. The doctor can retreat and re-insert the needle at the same insertion point and the cutting force exerted on the needle tip will be null. This algorithm allows stop without any dissipative element, also it permits to replicate the small movement of rejection that occurs when the needle stops in the body.

For the implementation of tracking wall, two parameters of wall position are necessary: ω_0 and ω_n. The position ω_0 establishes the edge of the wall, this is fixed and it cannot be updated as this is the set point of the virtual environment. So, when the needle is inserted for the first time ω_0 and ω_n are equals. ω_n is the position of the mobile wall which follows the needle displacement. This position is updated depending on tool displacement through the limb. To update ω_n equation 3 is used, and this process is declared as *"Update with movement"* in the flow diagram of Fig. 4.

$$\omega_n = \begin{cases} x_t - \Delta_{new} & \text{if } x_t > \omega_0 \\ \omega_0 & \text{if } x_t \leq \omega_0 \end{cases} \tag{3}$$

x_t is the current position of the needle tip. Δ_{new} is a variable called *"safety position"*, that computes the maximum distance required to reproduce the desired force, and the separation between the current point and ω_n while the tool is moving.

$$\Delta_{new} = \frac{f_R}{K_t} \tag{4}$$

Where f_R is the desired force to reproduce the cutting force and K_t is a tuning parameter that allows stabilising the value of the *"safety position"*. So, it should be calculated for each layer according to the desired forces. As a first assumption, the desired forces are taken from force profiles previously obtained [6].

When the tool stops inside the body, the position ω_n must be computed at each iteration. To achieve this, a variable m is calculated by (5). It represents represent the slope of the line defined by the stop point and ω_n.

$$m = \frac{x_t - \omega_n}{n} \tag{5}$$

Once m is defined, its value remains unchanged during the rest of the process. The update of ω_n at each iteration is given by (6).

$$\omega_n = \begin{cases} m * i + \omega_{n-1} & \text{if } i \leq n \\ \omega_n = \omega_{n-1} & \text{if } i > n \end{cases} \tag{6}$$

Where n is the total number of iterations to update ω_n, and i is the number of iteration computing to equal ω_n respecting the position where the stop occurs. In the case of n, this value must be tuned as it depends on the controller performances. In our tests, if the value of n is greater than $10 \cdot 10^6$, when the tool stops, due to the number of iterations the tool is rejected just a small distance as the system is not able to update ω_n on time. This behavior is desired as this is a phenomenon that also occurs when the needle is rejected by the muscles. This process, that involves the computation of m and the update of ω_n is called in the flow diagram *"Update with stop"*.

The purpose of this algorithm is to move the wall in front of the needle, so the next time the user moves the needle, he will face a wall as he found it when the virtual limb was penetrated the first time. The force exerted by the system (f_e) is calculated using (7):

$$f_e = \begin{cases} K_t(\omega_n - x_t) = K_t(\Delta_h) & \text{if } x_t \geq \omega_0 \\ 0 & \text{if } x_t < \omega_0 \\ 0 & \text{if } x_t < x_{t0} \end{cases} \tag{7}$$

where x_{t0} is the position where the tool stops inside the body. With these parameters, it is now possible to detail the operation of the tracking wall algorithm. Fig. 4 shows

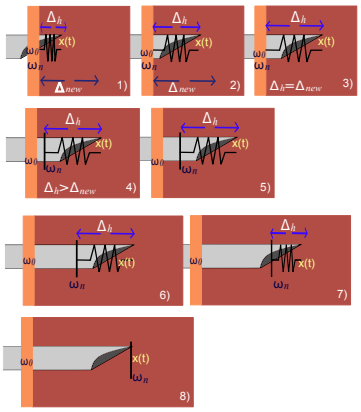

Figure 3. Evolution of a needle insertion, step by step, with the state of the virtual spring.

the flow diagram of the algorithm. The steps are as follows:

1. Initial conditions: $\omega_n = \omega_0$.
2. When the tool overpassed ω_0, the force exerted is calculated using (7) and is updated in each iteration of the algorithm.
3. The tool continues its trajectory through the virtual body. The algorithm compares Δ_h against Δ_{new}.
4. If $\Delta_h > \Delta_{new}$, it means that the tool has overpassed the "safety position", also the system is exerting the desired force for this layer
5. The tool continues its trajectory and now the value ω_n is updated using (3).
6. When the system detects that the tool velocity is zero ($\dot{x}_t = 0$), meaning that the user stopped its motion. The position is saved as x_{t0}, and the algorithm computes ω_n to update the wall using (6), while the speed remains in null.
7. When the user moves again the tool, the algorithm compares the actual position x_t with the value x_{t0} saved when the stop occurred. The value x_{t0} helps to memorize the position of the needle.
8. if $x_t < x_{t0}$, the force exerted is null, that means that the tool is retreated and then moved forward but not enough to start to cut new skin or muscle, but if $x_t > x_{t0}$ the force is calculated and the process restart from step 2.

Fig. 3 shows a graphical representation of the algorithm steps, where each number of the figure correspond to each number of the step described aforementioned.

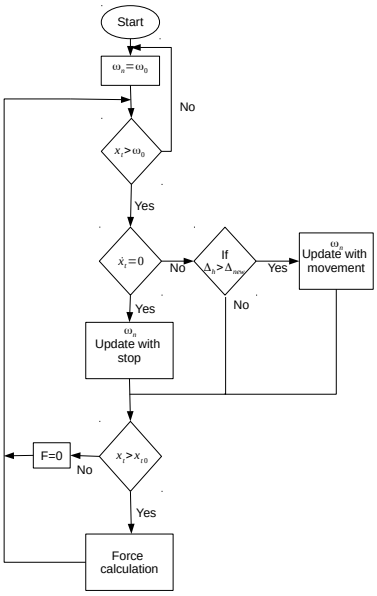

Figure 4. Flow Diagram of Tracking Wall algorithm.

3 Experimental results

To validate this algorithm, a Virtuose™ 6D Desktop, is used. The algorithm is implemented in Matlab Simulink®, using the library developed by the provider.

3.1 Needle insertion

The force profile used to test tracking wall were obtained from the experiments made by [6]. In this paper authors, performed a needle insertion, and got the cutting force and the thickness between layers, obtaining a graphic of force and displacements. The parameters taken from this test which are applied on tracking wall method are, the cutting force 1 N and the thickness of the layer 12 mm. With these parameters it is possible to tune K_t to calculate Δ_{new}. The stiffness value is $K_t = 500$ N/m (tuned value), the desired force is 1 N and the $\Delta_{new} = 2$ mm that is the value calculated after applied (4) and the wall (virtual set point) is located in $\omega_n = 0$ mm.

Fig.5 shows the results obtained along 1-DoF using the values aforementioned. In this test the needle is in free motion from -50 mm to 0, therefore the value of ω_n and ω_0 remains null.

Once the needle gets into the pink zone (virtual shoulder) the force is computed raising his value softly as long the needle is moving, reaching the desired force from $t = 0.7$s to $t = 1.3$s. This test helps to see how the algorithm updates the position of ω_n.

3.2 Needle insertion with a stop

The second experiment concerns the response of the wall and the behavior of the force when a stop occurs inside the virtual limb. As can be seen in Fig. 6 this test uses the same parameter of K_t , Δ_{new} and the desired force, setting $\omega_0 = 15$ cm. When the needle reaches the blue zone, the

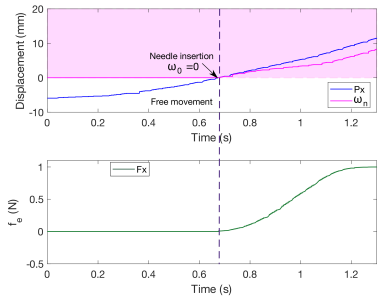

Figure 5. First stage of needle insertion (1 dof)

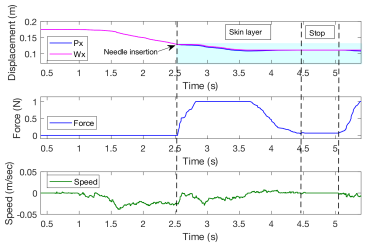

Figure 6. Tracking wall in 1-DoF with stops inside the body

force begins to increase until the desired value which is reached $t = 2.8$s. At $t = 4.5$s the needle stops, and speed and force decreased to zero. Then at $t = 5.3$ s, the needle started again its movement and the force begins to increase gradually as well as the speed.

3.3 Stop and backward motion

Fig. 7 shows the behavior of the tool when a stop occurred inside the body. In the zoomed area one can observe the distance between the actual position and the ω_n position, that distance is the value of $\Delta_{new} = 2$ mm, thus when the user stops within the body, m is computed updating the value of ω_n to equal the actual position and decrementing the force value. However, in our case, we search to replicate the needle rejection motion when the user stops within the body. This is due to the muscles behavior. Therefore, m is computed using $n > 10 \cdot 10^6$ in order to obtain the desired behavior. Afterward, the tool is moved back and then moved forward. As the needle tip cut previously the same tissue, no force is applied until the tool position goes beyond the saved position.

3.4 Tracking wall in 3-DoF

A fourth test is performed in a 3-DoF case. For this test, a virtual shoulder is represented as a hemisphere with a radius of 0.15m (Fig (8)). In this virtual shoulder, nine inner layers represent the different tissues and the limits of it. The desired forces for each layer are 1.5 N, 1 N, 1.8 N, 2 N, 3 N, 3.5N, 2.5 N, 4 N, and 4.5 N respectively. These values are arbitrarily chosen and are easily customized to reproduce different patients. The first layer represents the contact zone (pink shaded area Fig. 9) that helps the user

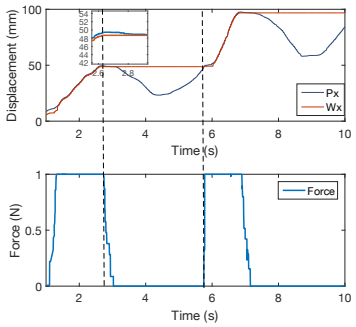

Figure 7. Illustration of the effect of stop and backward motions.

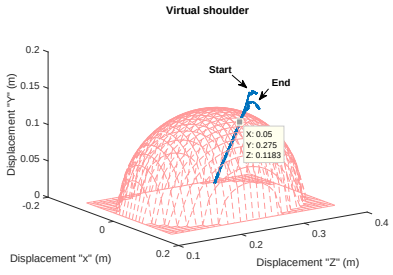

Figure 8. Virtual shoulder and trajectory

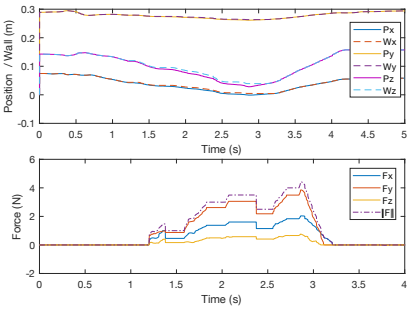

Figure 9. Tracking wall in 3-DoF: Positions and forces in axis X, Y and Z.

to feel the borders of it. The user must apply more force as the needle penetrates in each layer. When the user reaches the ninth layer the needle is pulled off from the virtual environment following the same trajectory as it was inserted.

To avoid any direction modification, a virtual fixture is implemented as proposes in [14]. It allows keeping the initial inclination in applying radial forces on the needle.

The results are shown in Fig. 9 where it can be appreciated the needle displacement in Px, Py, Pz with their corresponding tracking walls ω_{nx}, ω_{ny} ω_{nz} and the desired forces in x,y, and z. As can be seen, the force magnitude is computed to corroborate the desired forces (purple dot line).

In this test the user is able to feel the different stiffness within the shoulder, as long as his movement gets deeper, the forces change gradually allowing him to feel the behavior of the force.

4 CONCLUSION

The proposed method, called Tracking Wall, helps to reproduce the desired cutting force that is exerted on the needle tip. This method, avoid the use of any damper to dissipate the energy when the user stops his movement, instead of it, the parameter ω_n is computed which is the position of the wall that tracks the position of the virtual tool. Using this wall, the user can perform stops inside the limb and the rejection will be minimal, instead of the full trajectory as it happens with spring and dampers approaches. As future work, a campaign of measurement will be performed with the participation of experienced physicians and medical students. The objective is to measure how much this method is close to real life applications and evaluate how much it can help to improve their skills.

ACKNOWLEDGMENT

The authors would like to thank the ANR (French National Research Agency) for financing SAMSEI project (ANR-11-IDFI-0034) under the supervision of Pr. X. Martin.

References

[1] Y. Kurita, H. Ohtsuka, K. Nagata, T. Tsuji, *Haptic rendering of a needle insertion by enhancing the real force response of a base object*, in *Haptics Symposium 2014, HAPTICS 2014* (2014), pp. 357–360, ISBN 9781479931316, ISSN 23247355

[2] B. Gonenc, H. Gurocak, *Haptic interface with hybrid actuator for virtual needle insertion and tissue cutting*, in *Haptics Symposium 2012, HAPTICS 2012 - Proceedings* (2012), pp. 451–455, ISBN 9781467308090

[3] E.K. Tveitå, R. Tariq, S. Sesseng, N.G. Juel, E. Bautz-Holter, *Hydrodilatation, corticosteroids and adhesive capsulitis: A randomized controlled trial*, in *BMC Musculoskeletal Disorders* (2008), Vol. 9, p. 53, ISBN 1471-2474, ISSN 1471-2474, http://bmcmusculoskeletdisord.biomedcentral.com/articles/10.1186/1471-2474-9-53

[4] M. Martinez, G.R. Doulatram, Essentials of Interventional Techniques in Managing Chronic Pain pp. 471–479 (2018)

[5] J.C. Granry, M.C. Moll, Tech. rep., Haute Autorité de la Santé (HAS) (2012)

[6] H. Kataoka, T. Washio, K. Chinzei, K. Mizuhara, C. Simone, A.M. Okamura, *Measurement of the Tip and Friction Force Acting on a Needle during Penetration*, in *Lecture Notes and Computer Science* (2002), pp. 216–223, ISBN 3540442243, ISSN 16113349, http://link.springer.com/10.1007/3-540-45786-0{_}27

[7] C. Zilles, J. Salisbury, *A constraint-based god-object method for haptic display*, in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots* (1995), Vol. 3, pp. 146–151, ISBN 0-8186-7108-4

[8] H. Courtecuisse, H. Jung, J. Allard, C. Duriez, Y. Lee, S. Cotin, *GPU-based Real-Time Soft Tissue Deformation with Cutting and Haptic Feedback. GPU-based Real-Time Soft Tissue Deformation with Cutting and Haptic Feedback GPU-based Real-Time Soft Tissue Deformation with Cutting and Haptic Feedback*, in *Special Issue on Biomechanical Modelling of Soft Tissue Motion* (Elsevier, 2010), Vol. 103, https://hal.inria.fr/hal-00686056

[9] D. Zerbato, P. Fiorini, *A unified representation to interact with simulated deformable objects in virtual environments*, in *Proceedings - IEEE International Conference on Robotics and Automation* (2016), Vol. 2016-June, pp. 2710–2717, ISBN 9781467380263, ISSN 10504729

[10] D. Lobo, M. Sarac, M. Verschoor, M. Solazzi, A. Frisoli, M.A. Otaduy, *Proxy-based haptic rendering for underactuated haptic devices*, in *2017 IEEE World Haptics Conference, WHC 2017* (2017), pp. 48–53, ISBN 9781509014255

[11] A. Hernansanz, D. Zerbato, L. Gasperotti, M. Scandola, P. Fiorini, A. Casals, *Improving the development of surgical skills with virtual fixtures in simulation*, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2012), Vol. 7330 LNCS, pp. 157–166, ISBN 9783642306174, ISSN 03029743

[12] H. Li, K. Kawashima, *Bilateral teleoperation with delayed force feedback using time domain passivity controller*, in *Robotics and Computer-Integrated Manufacturing* (2016), ISSN 07365845

[13] K.J. Kuchenbecker, J. Fiene, G. Niemeyer, *Improving contact realism through event-based haptic feedback*, in *IEEE Transactions on Visualization and Computer Graphics* (2006), Vol. 12, pp. 219–229, ISBN 1077-2626 VO - 12, ISSN 10772626

[14] R. Kikuuwe, N. Takesue, H. Fujimoto, *Passive virtual fixtures based on simulated position-dependent anisotropic plasticity*, in *Proceedings - IEEE International Conference on Robotics and Automation* (2007), pp. 3263–3268, ISBN 1424406021, ISSN 10504729