Energy-Based 6-DOF Penetration Depth Computation for Penalty-Based Haptic Rendering Algorithms

Maxim Kolesnikov and Miloš Žefran

Abstract—Existing penalty-based haptic rendering approaches compute penetration depth in strictly translational sense and cannot properly take object rotation into account. We aim to provide a theoretical foundation for computing the penetration depth on the group of rigid-body motions $SE(3)$. We propose a penalty-based six-degrees-of-freedom (6-DOF) haptic rendering algorithm based on determining the closest-point projection of the inadmissible configuration onto the set of admissible configurations. Energy is used to define the metric on the configuration space. Once the projection is found the 6-DOF wrench can be computed. The configuration space is locally represented with exponential coordinates to make the algorithm more efficient. Numerical examples compare the proposed algorithm with the existing approaches and show its advantages.

I. INTRODUCTION

Six-degrees-of-freedom (6-DOF) force rendering algorithms are an active research area. Researchers are trying to come up with a good haptic rendering algorithm both in terms of realism as well as stability of haptic rendering and no 'standard' 6-DOF force rendering algorithm has yet emerged.

Our work focuses on the formulation of a new theoretically justified, physics-based approach to penetration depth computation for 6-DOF penalty-based haptic rendering algorithms. The algorithm we propose is essentially a penalty-based method in which contact forces are computed as a function of current and desired configurations of a rigid body. Similarly to other approaches, the desired configuration is the configuration that takes a rigid body out of the collision. However, while most other approaches only consider translational, but also rotational distance. To the best of our knowledge, all existing attempts to define a generalized penetration depth (e.g. [10]) use the metric based on strictly translational distance.

The paper is organized as follows: related previous work is discussed in Section II; in Section III we describe our haptic rendering algorithm; in Section IV we discuss implementation-related issues; and in Section V we present some numerical examples comparing the proposed ideas to the existing approaches. We conclude the paper with the discussion in Section VI.

II. RELATED WORK

The most common approach to the computation of contact forces in a typical 3-DOF display involves penalty-based methods [1]. Once a collision between rigid bodies is detected, penetration depth is estimated. Penetration depth is defined in this case as the minimum translational distance to separate the two bodies. Estimated penetration depth is used to compute the magnitude of the contact force. This approach is also known as direct rendering. Its main advantage is that the dynamics of the object entirely depends on the user of the haptic device so there is no need to solve for object’s dynamic behavior. The disadvantage is the possible instability as a result of penetration depth values getting too large and discontinuities in the displayed force.

Similar approach can be used for penalty-based 6-DOF haptic rendering [2]–[5]. Contact torques in this case are computed as a cross product between the vector from the center of mass of the object to the contact point and the force vector. To improve stability, this approach can be modified [6] by introducing a model of the object to which the calculated forces and torques are applied. This model interacts with the real haptic device through virtual coupling [7].

In contrast, constraint-based methods [8] of contact force computation are based on the notion of a substitute virtual object. Its dynamics is being constantly updated. Feedback force is calculated so that the user-controlled object is dragged towards the substitute virtual object. These methods can produce more accurate results at the expense of increased computational complexity. However, some recent results [9] promise to overcome some limitations of constraint-based methods.

Impulse-based techniques [5] implement the contact between two rigid bodies as a series of micro-collisions, and impulses are applied in order to prevent interpenetration. These techniques produce visually acceptable but haptically unconvincing results [5]. They can be used in combination with penalty-based methods [5].

Our work is directly related to the penalty-based 6-DOF haptic rendering approach. We propose a new method to compute the penetration depth which accounts not only translational, but also rotational distance. To the best of our knowledge, all existing attempts to define a generalized penetration depth (e.g. [10]) use the metric based on strictly translational distance.

III. PROPOSED ALGORITHM

A. Formal Description of Rigid Body Motion

We follow the approach in [11] for kinematic and dynamic modeling. For a shorter treatment the reader can also consult
where \( \mathbb{SO}(3) = \{ R \in \mathbb{R}^{3 \times 3} | RR^T = I_{3 \times 3}, \det R = 1 \} \) is the special orthogonal group in \( \mathbb{R}^3 \) [11]. Both \( \mathbb{SO}(3) \) and \( \mathbb{SE}(3) \) are Lie groups [13].

At the identity element of a Lie group the tangent space has the structure of a Lie algebra which completely captures the local structure of the group. The Lie algebra of \( \mathbb{SO}(3) \) is given by \( \mathfrak{so}(3) = \{ \Omega \in \mathbb{R}^{3 \times 3} | \Omega^T = -\Omega \} \) [11]. Each element \( \Omega \in \mathfrak{so}(3) \) can be mapped onto a vector \( \omega \in \mathbb{R}^3 \) so that \( \Omega x = \omega \times x \), where the symbol \( \times \) denotes the cross product of two vectors. We will also write \( \Omega = \tilde{\omega} \).

Similarly, the Lie algebra of \( \mathbb{SE}(3) \) is given by [11]

\[
\mathfrak{se}(3) = \left\{ \mathfrak{X} = \begin{bmatrix} \Omega & v \\ 0_{1 \times 3} & 0 \end{bmatrix} \mid \Omega \in \mathfrak{so}(3), v \in \mathbb{R}^3 \right\},
\]

(2)

An element of \( \mathfrak{se}(3) \) is referred to as a \textit{twist}. Here each twist can also be mapped onto a vector \( \xi = [v \quad \omega]^T \in \mathbb{R}^6 \) where \( \omega \in \mathbb{R}^3 \) is given by \( \Omega = \tilde{\omega} \). Vector \( \xi \in \mathbb{R}^6 \) represents the \textit{twist coordinates} of the twist \( \mathfrak{X} \), and we write \( \mathfrak{X} = \tilde{\xi} \).

Every possible rigid body configuration is represented by an element \( g \) of \( \mathbb{SE}(3) \) which in turn can be locally represented by a \( 6 \times 1 \) vector \( \tilde{\xi} \) of \textit{exponential coordinates} through the (matrix) exponential mapping, \( g = e^{\tilde{\xi}}, \tilde{\xi} \in \mathfrak{se}(3) \) [11].

\section*{B. Algorithm Description}

Consider a system comprised of rigid bodies \( A \) and \( B \) (Fig. 1). The rigid body \( A \) is controlled by the user with the haptic device and the rigid body \( B \) is an obstacle in the virtual environment. Let us denote the configuration space of the system by \( Q \). To keep things simple we only consider the relative position of \( A \) with respect to \( B \), and we assume that there are no boundary surfaces so that the configuration space is \( Q = \mathbb{SE}(3) \). Let \( A(q) \) be the subset of \( \mathbb{R}^3 \) occupied by the object \( A \) at configuration \( q \in Q \). Then the admissible configuration set \( C \) is given by a set of all possible configurations \( q \in Q \) for which the two bodies do not collide:

\[
C = \{ q \in Q | A(q) \cap B = \emptyset \}.
\]

(3)

Consider the situation when \( A \) is in collision with \( B \). We denote this inadmissible configuration by \( p \). Assuming that the admissible set \( C \subset Q \) is nonempty we can define the distance function [14] as

\[
d_C(p) = \min \{ d(p, q) \mid q \in C \},
\]

(4)

which returns the distance between any inadmissible configuration \( p \) and the admissible set \( C \). Here the function \( d(p, q) \) is the \textit{distance metric function} [15] between configurations \( p \) and \( q \).

For every inadmissible configuration \( p \) we can find an optimal admissible configuration \( q \) such that \( q \in P_C(p) \), where

\[
P_C(p) = \{ q \in \overline{C} \mid d_C(p) = d(p, q) \},
\]

(5)

and \( \overline{C} \) is the closure of \( C \). The configuration \( q \) is called a \textit{closest-point projection} of \( p \) onto \( C \) [14].

Naturally, the question comes up of what distance metric function to choose for the computation of \( d(p, q) \). The metric should be such that the closest point projection is not affected by scaling of the objects. This is important because if, e.g., an object \( Q \) is a scaled version of another object \( P \), the two objects would look identical to the user (in appropriate views) so the computed desired configurations \( q_P \) and \( q_Q \) should also look the same (i.e., the translational part of \( q_Q \) should be the scaled version of that of \( q_P \)). This condition rules out e.g. \( d(p, q) = d(I_{3 \times 4}, r^{-1}q) = \| \tilde{\xi}_{r^{-1}q} \|_2 \) as a possible solution, where \( \tilde{\xi}_{r^{-1}q} \) represents the exponential coordinates of the configuration \( r^{-1}q \).

An attractive and physically meaningful solution is to define the distance metric as the kinetic energy needed to move between configurations \( p \) and \( q \) in a unit of time:

\[
d(p, q) = T(p, q) = \tilde{V}^T M \tilde{V},
\]

(6)

where \( \tilde{V} \) is defined below and roughly corresponds to the twist coordinates of the body velocity of the rigid body \( A \) moving from \( p \) to \( q \), while \( M \) is the generalized inertia matrix.

To compute the body velocity \( \tilde{V} \) represented in twist coordinates we can use the fact that if \( g(t) \in \mathbb{SE}(3) \) describes the motion of a rigid body, then the body velocity \( \tilde{V} \in \mathfrak{se}(3) \) of the rigid body is [11]

\[
\tilde{V} = g^{-1} \frac{dg}{dt}.
\]

(7)

To compute \( \tilde{V} \), take

\[
g \approx p \exp \left( \left[ \frac{\tilde{\xi}_{r^{-1}q}}{2} \right] \right),
\]

(8)

the configuration located at the midpoint between \( p \) and \( q \). The derivative of \( g \) can be approximated by

\[
\frac{dg}{dt} = q - p \quad \frac{\Delta t}{\Delta t}.
\]

(9)

The time interval \( \Delta t \) is assumed to be roughly the same for each possible configuration \( q \) considered. Therefore, since we are only interested in the value of the derivative computed for some admissible configuration \( q_1 \) relative to all other possible admissible configurations \( q_2, q_3, \ldots \), assume for simplicity \( \Delta t = 1 \) and \( \tilde{V} \approx q - p \). We thus have the final expression:

\[
\left[ \tilde{V} \right] \approx \exp \left( \left[ -\frac{\tilde{\xi}_{r^{-1}q}}{2} \right] \right) p^{-1} (q - p).
\]

(10)
Finally, \( \tilde{V} \) are the twist coordinates of \( \tilde{V} \wedge \).

Recall that the generalized inertia matrix \( M \) in the case where the body reference frame is placed at the center of mass of a rigid body is given by

\[
M = \begin{bmatrix}
ml I_{3\times3} & 0_{3\times3} \\
0_{3\times3} & H
\end{bmatrix},
\]

(11)

where \( m \) is the mass of the object and \( H \) is the inertia tensor.

The distance metric defined through Eqs. (6) and (10) is insensitive to scaling of the objects. An important property of this distance metric is its bi-invariance. That means that no matter how the inertial frame and the body reference frame are chosen, the distance metric \( d(p, q) \) evaluates to the same result. In other words \( d(p, q) = d(ap, aq) = d(pb, qb) \) for arbitrary \( a, b \in SE(3) \). Bi-invariance of the distance metric is sufficient for the metric to be well-defined [15]. Therefore, there is no need to stay in the neighborhood of the identity element of \( SE(3) \) when measuring the distance between two configurations. It is also worth mentioning that the mass properties of the object will not affect the metric, but it does depend on the shape of the object. The latter can be seen as an advantage over the approach in [2] where the reaction wrench is independent of the shape of the object.

C. Wrench Computation

Suppose we determine the optimal admissible configuration \( q \) as the closest-point projection onto the admissible set \( C \) in accordance with (5). Then the penalty-based approach for wrench computation can be used. First we compute vector \( \varepsilon \) of the differences between the twist coordinates of the (current) inadmissible configuration \( p \) and the optimal admissible configuration \( q \). Note that in order for the result to be independent of the chosen inertial reference frame, we compute the difference at the identity element of \( SE(3) \):

\[
\varepsilon = \xi_{p^{-1}q} - \xi_{p^{-1}p} = \xi_{q^{-1}q}.
\]

(12)

Vector \( \varepsilon \) is equal to zero if and only if \( p = q \). This can only happen when \( p \in C \) which means that the rigid body is not in collision.

Once the penetration has been computed, any appropriate scheme can be used to compute the wrench. Motivated by [2] we use the proportional-derivative (PD) control law. Accordingly, the wrench \( W(t) = \begin{bmatrix} f(t) & \tau(t) \end{bmatrix}^T \) at each time instance is computed as

\[
W(t) = K_P \varepsilon(t) + K_D \frac{d\varepsilon(t)}{dt},
\]

(13)

where \( K_P \) and \( K_D \) are \( 6 \times 6 \) matrices. In fact, matrices \( K_P \) and \( K_D \) have the same physical meaning as stiffness and damping matrices, respectively. Even though the stiffness matrix is in general asymmetric [16], for the sake of simplicity one can assume that it is symmetric and diagonal in the form of \( k_P h_{6\times6} \) so that a single stiffness parameter \( k_P \) is needed. The same simplification can be made for matrix \( K_D \).

IV. IMPLEMENTATION

A. Overview

The procedure for computation of the six-dimensional wrench can be summarized as follows:

1) Check for collision. If there is no collision, continue simulation and go to Step 1 in the next time instance, otherwise save the current inadmissible configuration in \( p \).

2) Using the proposed kinetic energy metric, find the admissible configuration \( q = \arg\min_{r \in C} d(p, r) \).

3) Compute the configuration matrix \( p^{-1}q \) corresponding to the relative displacement between \( p \) and \( q \).

4) Compute the difference vector \( \varepsilon = f_1(p^{-1}q) \) using Eq. (12).

5) Compute the body wrench \( W_b = f_2(\varepsilon) \) using Eq. (13) and find the corresponding spatial wrench \( W_\Sigma \).

6) Send the wrench \( W_\Sigma \) to the controller of the haptic device.

7) Go to Step 1 in the next time instance.

In order to find the closest-point projection onto the admissible set \( C \) in Step 2 it is necessary to minimize the function \( d(p, q) \). There are two different ways of approaching this problem: a numerical optimization and analytical approximation methods.

It turns out that the numerical optimization is not fast enough. Even simple numerical examples take several seconds to run in Matlab on a 1 GHz CPU using conjugate gradient search method. While this time can be reduced through compilation and by using a faster CPU it is clear that the numerical optimization is not appropriate for a real-time implementation.

B. Analytical Solution – Two-Dimensional Case

As an alternative to the numerical optimization we propose an analytical approximation of the solution that is fairly accurate and can be computed at sufficiently high rates. We first describe the method for a two-dimensional case, where the analytical solution is exact. Then we discuss how to use these results in a general three-dimensional case.

In a two-dimensional case a planar body can only move on a plane and rotate around the axis perpendicular to the plane. Let us assume that the body is a convex polygon. Let us also assume that only one vertex of the body is in collision. This is virtually always the case for real-time simulations with refresh rates of several hundred hertz. After collision is detected the body resides in some inadmissible configuration \( p \). Such situation is depicted in Fig. 2, where the region
below the x-axis coincides with the planar obstacle. Here, T is the body reference frame placed at the center of mass of the body, h is the y-coordinate of the reference frame T, and \( \varphi \) is the angle between the reference frame T and the main spatial reference frame.

We are looking for the admissible configuration \( q \in P_c(p) \) of the body that is a closest-point projection of the configuration p onto the admissible set C. Such a configuration \( q \) minimizes the distance metric (6). The configuration \( q \) (Fig. 3) can be parametrized by the y-coordinate \( \beta \) of the body reference frame T and the angle \( \theta \) between the reference frame T and the main spatial reference frame. It can be shown that the x-coordinate of the reference frame T corresponding to the optimal configuration \( q \) is the same as the x-coordinate of \( p \).

In this general setup it can be shown that Eq. (6) results in

\[
T = m(\beta - h)^2 + 4H_{3,3} \sin^2 \left( \frac{\theta - \varphi}{2} \right),
\]

where \( H_{3,3} \) is the (3,3) element of the inertia tensor matrix \( H \). Given parameters \( h \) and \( \varphi \), we can find \( \beta \) and \( \theta \) such that (14) is minimized.

It is easy to see that in the optimal configuration, the y-coordinate of the vertex \( v \) has to equal 0. This results in

\[
\beta = -v_x \sin \theta - v_y \cos \theta.
\]

Furthermore, to prevent situations like the one depicted in Fig. 4, we also need to satisfy the constraint

\[
0 \leq \theta \leq \pi - \gamma,
\]

where \( \gamma \) is the internal angle of the planar body associated with the vertex \( v \). This constraint will be used later to check whether the obtained solution is in the admissible configuration set.

Using (14) and (15) our optimization function is

\[
J = m(-v_x \sin \theta - v_y \cos \theta - h)^2 + 4H_{3,3} \sin^2 \left( \frac{\theta - \varphi}{2} \right).
\]

The critical point \( \tilde{\theta} \) of the functional (17) is the solution of

\[
\frac{\partial J}{\partial \theta} \bigg|_{\theta = \tilde{\theta}} = m(v_x^2 - v_y^2) \sin 2\theta + 2mv_x v_y \cos 2\theta + 2mh(v_x \cos \theta - v_y \sin \theta) + 2H_{3,3} \sin(\theta - \varphi) = 0.
\]

It can be shown that solving (18) for \( \tilde{\theta} \) results in

\[
\tilde{\theta} = 2\arctan x,
\]

where \( x \) is a solution of a quartic equation

\[
A_1 x^4 + B_1 x^3 + C_1 x^2 + D_1 x + E_1 = 0,
\]

with the coefficients

\[
A_1 = mv_x v_y - mv_y h + H_{3,3} \sin \varphi,
B_1 = 2(H_{3,3} \cos \varphi - mh v_y - mv_x^2 + mv_y^2),
C_1 = -6mv_x v_y,
D_1 = 2(H_{3,3} \cos \varphi - mh v_y + mv_x^2 - mv_y^2),
E_1 = mv_x v_y + mv_y h - H_{3,3} \sin \varphi.
\]

Closed form solutions of quartic equations can be obtained using for example classic Ferrari’s method [17].

After solving Eq. (20) for \( x \) and obtaining the critical value \( \tilde{\theta} \) of \( \theta \) the constraint (16) has to be checked and the value \( T_{\theta=\tilde{\theta}} \) of the energy function (14) has to be computed. Also, the border values \( T_{\theta=0} \) and \( T_{\theta=\pi-\gamma} \) of the energy function have to be computed for \( \theta = 0 \) and \( \theta = \pi - \gamma \). If the obtained \( \tilde{\theta} \) satisfies the constraint, then the optimal value \( \theta_0 \) of \( \theta \) is

\[
\theta_0 = \arg \min_{\theta} \{ T_{\theta=0}, T_{\theta=\tilde{\theta}}, T_{\theta=\pi-\gamma} \}.
\]

If the obtained \( \tilde{\theta} \) does not satisfy the constraint (16), then

\[
\theta_0 = \arg \min_{\theta} \{ T_{\theta=0}, T_{\theta=\pi-\gamma} \}.
\]

C. Analytical Approximation – Three-Dimensional Case

The main idea of the proposed method is to obtain a solution for a 3D case by exploiting the analytical solution for the 2D case. If we have a 3D virtual environment with a user-controlled convex rigid body A, then we can apply the 2D technique described above in Section IV-B to any cross section of the rigid body containing the vertex \( v \) in collision.

Without loss of generality let us arrange the axes \( X, Y \) and \( Z \) so that the plane \( XZ \) is the surface of the obstacle and some vertex \( v \) collided with it. The projection of the rigid body on the \( XY \)-plane thus corresponds to Fig. 2.

Now consider the family of planes \( \{ P_{\delta} \mid \delta \in [0, \pi] \} \) obtained by rotating the \( XY \) plane around the line parallel to the \( Y \)-axis and containing the vertex \( v \). Each plane \( P_{\delta} \) of the family is thus parametrized by its rotation angle, \( \delta = \angle(P_{\delta}, XY) \).

Each \( \delta \) generates a planar cross-section \( P_{\delta} \cap A \). For each cross-section, the 2D procedure in Section IV-B produces an
optimal solution $G_{2D}(\delta) = [\beta_\delta \ \theta_\delta]^T$. Recall that $\theta_\delta$ is the optimal angle of rotation in the plane $P_\delta$. It is easy to see that the corresponding axis of rotation $\omega_\delta$ expressed as a vector in 3D for a particular value of $\delta$ can be determined as $\omega_\delta = [\sin \delta \ 0 \ \cos \delta]^T$.

The 2D solution can be expressed in 3D by adding the information about the rotation axis $\omega_\delta$. Let us denote $G_{3D}(\delta, G_{2D}(\delta)) = [\beta_\delta \ \theta_\delta \ \omega_\delta]^T$. For each $G_{3D}(\delta, G_{2D}(\delta))$ there is a corresponding value of optimal 3D configuration $q(G_{3D}(\delta, G_{2D}(\delta)))$ and the distance metric $d(p, q(G_{3D}(\delta, G_{2D}(\delta))))$. Then the optimal values $[\beta_\delta, \omega_\delta, \theta_\delta]^T$ with respect to the distance metric are determined by

$$\delta_0 = \arg \min_{\delta \in [0, \pi]} d(p, q(G_{3D}(\delta, G_{2D}(\delta)))) \quad (28)$$

In practice one way to solve Eq. (28) is to discretize the domain of $\delta$ and thus work with an approximation such as

$$\delta_0 \approx \arg \min_{\delta \in \left\{ \frac{\pi}{N_\delta}, \frac{2\pi}{N_\delta}, \ldots, \frac{(N_\delta-1)\pi}{N_\delta} \right\}} d(p, q(G_{3D}(\delta, G_{2D}(\delta)))) \quad (29)$$

where $N_\delta$ is the number of discretization points. The approximation (29) is particularly appealing when multiple CPU cores are available. Another approach is to use some variation of gradient search method to solve Eq. (28).

D. Analytical Approximation – Performance

One of the most important requirements for any haptic rendering algorithm is its ability to provide results in the specified time interval. In haptics, 1 kHz is the de facto standard refresh rate. Studies have shown that the absolutely minimal acceptable haptic refresh rate is 500 Hz [18].

We have analyzed the performance of the proposed analytical method on a 1 GHz CPU. For the purpose of this analysis the proposed method can be divided into three stages:

1) Computation of a 2D solution. Query rate for this stage only was measured to be about $f_1 = 60$ kHz. It is roughly constant for all possible cases.
2) Computation of the angle $\gamma$ of each planar cross-section. Query rate for this stage in the worst case scenario was measured to be at least $f_2 = 226$ kHz. For other cases it was greater than this value.
3) Procedure for combining the family of 2D solutions into a 3D optimal solution. Query rate for this stage is very high compared to the other two stages, so this stage can generally be neglected.

Query rate for stages 1 and 2 combined is therefore $(f_1^{-1} + f_2^{-1})^{-1} = 47$ kHz.

The computation in stages 1 and 2 should be performed $N_\delta$ times, so, taking into account the overhead in stage 3, the effective query rate is $f_0 = \frac{1}{N_\delta}$. For realistic haptic rendering we need to make sure that $f_0 \geq 1$ kHz, therefore in our case $N_\delta \leq 47$. This would give us with the step size $\Delta \delta \geq 3.8^\circ$ which gives sufficient degree of accuracy.

Another major component to be accounted for in performance analysis is the collision detection routine. However, recent popularity of multicore CPUs allows to run a separate collision-detection thread on another core, effectively separating the two processes. Computations in Steps 3-5 of Sec. IV-A are typically very fast regardless of the choice of matrices $K_P$ and $K_D$.

V. NUMERICAL EXAMPLE

To illustrate the differences in the results obtained using our proposed method and traditional penalty-based computational approaches let us look at virtual simulation of a collision of two rigid bodies. Consider a virtual environment as in Example 1, where the cube can move freely in the half-space $\{y > 0\}$. The obstacle plane prevents the cube from moving into the other half-space. The initial inadmissible configuration $p$ is described by exponential coordinates $\xi_0 = [15.7174 \ 31.7380 \ 0 \ 0 \ 0.9196]^T$. We looked at the following scenario: initially, the cube is rotating with the body velocity $V = [0 \ -5 \ 0 \ 0 \ 0.2]^T$. Then, at a certain point it collides with the obstacle plane and haptic rendering algorithm is run to compute the required forces and torques. These computed values then in turn effect the dynamics of the cube until it gets out of the collision situation, at which point it continues to move with a constant velocity. The simulation was run with a sampling period of $\Delta T = 0.05$ sec for $N_t = 120$ steps.

We run our proposed method and the traditional penalty-based approach suitable for 6-DOF haptic rendering (as in [2]) side by side. As expected, they yield very different results. In Fig. 5 the projection of the trajectory of the center of mass on $XY$-plane is shown for each of the two cases. One can see that the trajectories diverge at a certain point during the simulation. This point is actually the instance when the collision is detected and haptic rendering algorithm is executed.

Rendered forces and torques are also very different for each of these two cases. For example, the force $f_1$ alongside $y$-direction and the torque $\tau$, in $XY$-plane are shown in Fig.
Fig. 6. Rendered force alongside y-direction using the proposed energy-optimal method (solid line) and the traditional approach (dashed line).

Fig. 7. Rendered torque in XY-plane using the proposed energy-optimal method (solid line) and the traditional approach (dashed line).

6 and 7. Results of this simulation show that the traditional penalty-based computation approach is definitely not optimal in terms of minimum energy metric.

VI. CONCLUSION

The proposed technique for 6-DOF penetration depth computation provides a theoretical basis for realistic haptic rendering of 6-DOF virtual environments. It is based on the analysis of the configuration matrix of the user-controlled haptic object and determining its energy optimal projection onto the set of admissible configurations. As we have shown, the configuration space allows for more accurate analysis of rigid body motion.

Solution for two-dimensional case can be obtained in closed form. By bisecting the object with a family of properly constructed planes, a family of two-dimensional solutions can be obtained from which the solution to the original three-dimensional problem can be deduced. Proposed analytical approximation is quite fast compared to numerical optimization-based techniques and its accuracy can be adjusted to take advantage of the available computational power.

The method can be extended to the case of N vertices in collision by superimposing the feedback wrenches \( W_1, W_2, \ldots, W_N \) computed for each of the \( N \) vertices. Other issues such as the ability to deal with the pop-through artifacts and friction can be dealt with in exactly the same way as done in existing penalty-based approaches [2], [3], [6].

In the future we plan to perform user studies and compare the performance of the proposed algorithm to the existing approaches from the perceptual point of view.

REFERENCES