Simulation of Soft Finger Contact Model with Rolling Effects in Point-Contact Haptic Interfaces

Gionata Salvietti[1], Monica Malvezzi[2], and Domenico Prattichizzo[1,2]

[1]Department of Advanced Robotics,
Istituto Italiano di Tecnologia, Via Morego 30, Genoa - Italy
gionata.salvietti@iit.it
[2]Department of Information Engineering and Mathematics,
Università degli Studi di Siena, Siena - Italy
{malvezzi, prattichizzo}@dii.unisi.it

Abstract. Computation of contact point trajectories and forces exchanged between two bodies in contact are relevant to several disciplines. The solutions proposed in the literature are often too complex to be implemented in real time simulations, especially if rolling effects are considered. In this chapter, an algorithm for fast simulation of soft-finger contact model with rolling effects is proposed. The main idea is to use Euler angle decomposition algorithm to quantitatively describe the torque exchanged about the normal at the contact point and the motion of the contact point due to rolling. The proposed algorithm is validated with simulations and a preliminary application to point-contact haptic interface is proposed.

Keywords: Contact Modeling, Soft Finger Model, Rolling Effects

1 Introduction

One of the key features of the human fingertips is the ability to resist moments, up to a torsional friction limit, about contact normal. Considering friction between the finger and the object, forces can be exerted in any direction that is within the friction cone for the contact and moreover, a torque about the normal to the contact plane can be applied [1]. This model is usually known in literature as *soft finger contact model* [2]. It is worth nothing that human fingers are actually surfaces, and manipulation of an object by a set of fingers involves *rolling* of the fingers along the object surface. Rolling can be defined as an angular motion between two bodies in contact about an axis parallel to their common tangent plane [3].

In this chapter, we took inspiration from the interaction finger/object to simulate a soft-finger contact model with rolling effects in Haptics. Algorithms for contact simulation needs computation efficiency to guarantee to the user

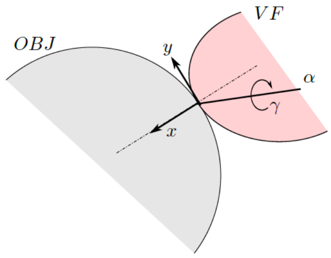


Fig. 1. Representation of the virtual finger touching a sphere

a correct perception of a stiff surface [5]. This makes usually intractable the real-time solution of classical equation describing rolling effect [1]. Another issue in simulating soft contact is represented in Fig. 1. The contact model assumes that torque is exerted only about the normal to the contact surface [4], [7], but in general, if only one finger is simulated to be in contact with an object, the exerted torque can be about a generic direction.

The contribution of this work is to extend the model of the soft-finger contact taking into account also possible rolling phenomena at the contact and keeping the same computational efficiency of point-based algorithm. The main idea is to use Euler angle decomposition algorithms to quantitatively describe the torque exchanged between the two bodies and the motion of the contact point due to rolling. We experimentally verified that our approach is more efficient in terms of computation time with respect to the integration of classical equation regulating rolling contacts. Moreover, we realized an application for Haptics, where we simulated the interaction between a fingertip and a plane.

The chapter is organized as it follows. In Section 2 the algorithm used is described in detail. In Section 3 validation of the method and experiment results on a simplified virtual environment are presented. Finally in Section 4, conclusion and future work are outlined.

2 Description of the Algorithm

The main target of this work is to find a computationally simple, but reliable algorithm for the simulation of the soft finger contact when rolling effects are considered. Although soft finger model implies a local deformation of the contact bodies, in this algorithm we do not evaluate the entity of such deformation, and the related stress distribution, and consider the contact bodies as undeformable. The effect of finger softness is represented only by the spin moment generated when the angular speed between the contact bodies have a component normal to the surfaces in the contact point.

The basic idea is to represent the relative displacement between contact bodies as a sequence of infinitesimal rotations. For each integration step, we use an Euler decomposition to get the rotational contribution along each axis of the reference frame built on the contact point.

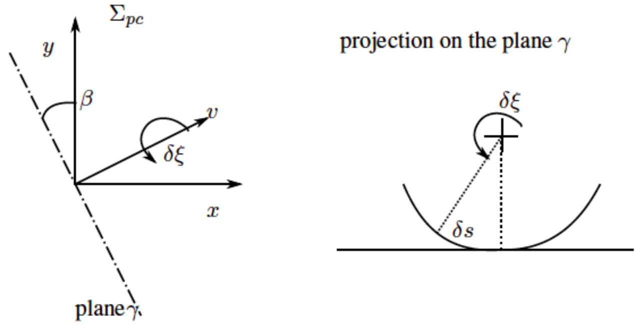


Fig. 2. Simplified model of the contact point evolution during rolling between a sphere and a plane with respect to a generic axis.

In the following we will assume that the contact forces fulfill the friction constraints, i.e. the contact force is inside the friction cone and then the contact surfaces have a relative rolling motion and no sliding is present at the contact point [9]. Let us assume that the relative rotation variation between the reference frames Σ_s and Σ_p can be represented by the Euler angles $\delta\phi$, $\delta\theta$ and $\delta\psi$, evaluated about the z, y and x axes, respectively.

The first component of the relative rotation $\delta\phi$ is adopted to evaluate the spin moment, that is reproduced by the haptic interface (HI)

$$\tau_n = k_z \delta\phi, \tag{1}$$

where k_z is a stiffness value that depends on surface properties and geometry.

The components $\delta\theta$ and $\delta\psi$ are employed to evaluate the contact point displacement, by means of the approximated method described below and shown in Fig. 2. Let γ be a plane, normal to the contact tangent plane, obtained by rotating the plane defined by the equation $x = 0$ by an angle β defined as

$$\beta = \mathrm{atan2}\left(\delta\theta, \delta\psi\right). \tag{2}$$

On each surface, the contact point will move along the intersection between the surface itself and the γ plane, the length of the contact point path is given by

$$\delta s = \frac{1}{k_\gamma}\delta\xi = \frac{1}{k_\gamma}\sqrt{\delta\theta^2 + \delta\psi^2} \tag{3}$$

1: **for** each integration step **do**
2: compute frames at contact point
3: compute rotation axis and angle
4: compute Rotation Matrix using Rodriguez's formula
5: compute Euler decomposition to get angles along contact frame $\delta\phi, \delta\theta, \delta\psi$
6: compute the Moment along the normal $\tau_n = k_z \delta\phi$
7: compute contact point position on the plane
8: update sphere position
9: **end for**

Algorithm 1: Computation of contact point position using Euler decomposition.

where k_γ is the relative curvature between the contact surfaces, evaluated with respect to the plane γ. For instance, if the contact surfaces are two spheres, with radii R_1 and R_2 respectively, the curvature is constant: $k_\gamma = \frac{1}{R_1} + \frac{1}{R_2}$. In the example analysed in this paper, shown in Fig. 2, in which a sphere is rolling on a flat surface, we have $k_\gamma = \frac{1}{R}$, where R is the sphere radius. The main steps of the proposed algorithm used to evaluate the relative motion between rolling surfaces are summarized in the Algorithm 1.

3 Application to Haptics

3.1 Validation of the algorithm

In order to validate our method, we compared the obtained trajectories of the contact points with those obtained by directly integrating the equation of rolling reported in [1]. For the integration, a four-five order Runge-Kutta algorithm was adopted (ODE45). All the simulations were realized using Matlab R2009a over a 2.4 GHz Intel Core 2 Duo, 4 GB RAM.

We simulated a sphere rolling on a plane. The Euler angles were computed starting from the rotation unit vector α and the rotation angle γ (Fig. 1) as described in the following. First, the rotation matrix $R(\alpha, \gamma)$ was computed using the Rodriguez formula [1]

$$R(\alpha, \gamma) = \alpha\alpha^T + (I(3 \times 3) - \alpha\alpha^T)\cos\gamma + S(\alpha)\sin\gamma, \tag{4}$$

where the superscript ^{T} means the transpose matrix and $S()$ is the skew operator. Then, the Euler angles were computed using the Roll-Pitch-Yaw convention [6] as

$$\delta\theta = atan2\left(\sqrt{r_{13}^2 + r_{23}^2}, r_{33}\right) \tag{5}$$

$$\delta\psi = atan2\left(\frac{r_{31}}{\sin\delta\theta}, \frac{r_{32}}{\sin\delta\theta}\right) \tag{6}$$

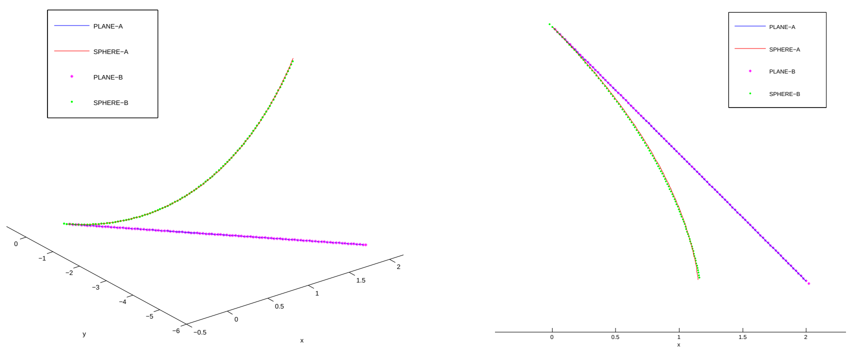
$$\delta\phi = atan2\left(\frac{r_{13}}{\sin\delta\theta}, \frac{-r_{23}}{\sin\delta\theta}\right), \tag{7}$$

where r_{ij} represents the components of the matrix $R(\alpha, \gamma)$. Note that the above representation has singularities in $\delta\theta = 0 \pm k\pi$. Moreover, equation (5) can be computed also as $\delta\theta = atan2(-\sqrt{r_{13}^2 + r_{23}^2}, r_{33})$. Anyway, we experimentally verified that this representation leads to a wrong approximation, so the solution proposed in (5) was always adopted.

In the sphere-plane simulation reported in Fig. 3 we considered, for each sampling step, $\delta\phi = 0.05$rad, $\delta\theta = 0.02$rad, $\delta\psi = 0.03$rad. The radius of the sphere was 10mm. The sampling time was 0.01s, the simulation time was 1s. The computation time necessary to perform the simulation using classic equations was 0.35s, while using the proposed approach was 0.01s. The maximum difference between the trajectories calculated with the two methods were 0.004mm, 0.004mm, 0.001mm along the plane reference system axis x, y, z respectively.

3.2 Experiment with CHAI 3D

The proposed algorithm has been exploited in a haptic rendering application, due to the low computational load and to the simplicity of the implementation. We used an Omega.6 HI [10] with three active translation and 3 passive rotation. For this reason, in this first experiment the moment exerted on the haptic pointer, due to the soft finger model adopted for the contact point, can not be reproduced by the HI. The value of the friction torque above the contact normal



(a) (b)

Fig. 3. (a) Trajectory of the contact point on the plane and on the sphere obtained with the classical formulation (A) and with the proposed algorithm (B); (b) Projection of the trajectories on the XY plane.

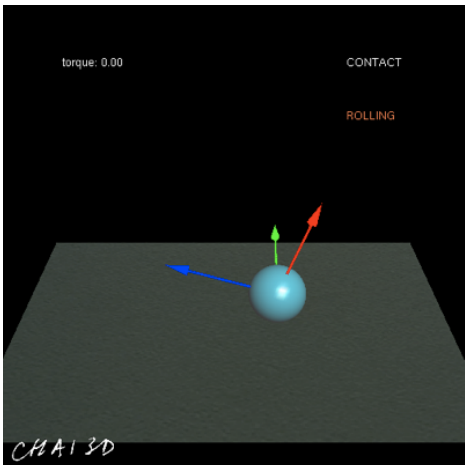


Fig. 4. Implementation with CHAI-3D. The white sphere represents the fingertip interacting with plane in the Virtual Reality.

direction is only visually shown. The whole application has been realized using the CHAI 3D library [11].

The contact detection and the force computation are important issues that have to be addressed during haptic simulations. We used a proxy-probe method [12] to generate the translational forces, while torque along the normal was computed as presented in Section 2. We reproduce the fingertip as a sphere and we simulate the interaction with a plane. The virtual reality scenario is shown in Fig. 4. Rolling conditions hold if the contact force comply with the friction constraints, and, in particular, if it is included in the friction cone. If the tangential component of the contact force exceed the friction limit, rolling conditions do not hold any more and the surfaces begin to slide each other. Different contact conditions have to be considered, and the contact mode transition is regulated by a Finite State Machine where the contact state can be one of the following: *rolling*, *contact breaking* or *sliding* [8]. The contact state transition can be predicted by the relative acceleration and contact forces that are computed from the measurements given by the HI. The condition are summarized as follows, where the subscripts R and S denote rolling and sliding contacts, N and T refer to the normal and tangential contact direction and μ_i is the coefficient of static friction and μ_{Si} is the coefficient of sliding friction and f if the force applied by the HI

- If $f_{i_N} = 0$ and $a_{i_N} > 0$, then contact is broken;
- If $f_{i_{NR}} > 0$, $\mu_i^2 f_{i_{NR}}^2 - f_{i_{TR}}^2 \geq 0$, $a_{i_{NR}} = 0$, $a_{i_{TR}} = 0$, then rolling is kept;
- If $f_{i_{NR}} > 0$, $\mu_i^2 f_{i_{NR}}^2 - f_{i_{TR}}^2 = 0$, $a_{i_{NR}} = 0$, $a_{i_{TR}}^2 \neq 0$, the rolling is switched to sliding;

- If $f_{i_{NS}} > 0$, $\mu_{Si}^2 f_{i_{NS}}^2 - f_{i_{TS}}^2 = 0$, $a_{i_{NS}} = 0$, then sliding is kept;
- If $f_{i_{NS}} > 0$, $\mu_{Si}^2 f_{i_{NS}}^2 - f_{i_{TS}}^2 \geq 0$, $a_{i_{NS}} = 0$, $v_{iTS} = 0$, then sliding is switched to rolling.

In a real scenario, if the fingertip is rolled over the plane the contact point moves both on the finger and the plane. If we implement a classic proxy-probe method, only the contact point on the virtual finger moves, while that on the plane remain fixed. Practically, no rolling effect is considered and the finger slips over the plane. This phenomenon is typical in HI where there are three active DoFs displaying translational forces and three passive DoFs measuring the orientation of the device end–effector, such as the Omega.6. To overcome this effect and increase the realism in the haptic experience, we considered a spring-damper connection between the position computed through our algorithm and the actual position of the probe. This solution leads to a contact point velocity different from zero also when rolling state is detected. We solved this issue allowing sliding condition only when a force tangent to the contact plane is exerted having an arbitrary module but the sign concordant with the sign of the displacement generated by the rolling. In fact, rolling condition would never be detected if a generic tangential force was enough to switch to sliding since to update the contact point position in rolling we consider the spring-damper connection between the actual and computed position of the probe. When sliding condition is detected, no spring-damper connection is considered.

Sliding condition can be detected also when the torque applied to the probe is such that the corresponding contact force exits from the friction cone. The threshold condition between rolling and sliding can be computed equaling the applied torque to the tangential force exerted, i.e

$$\Delta\theta \, r \, k_\tau = f_N \mu_s$$

where $\Delta\theta$ is the angular displacement measured by the HI, k_τ is the virtual stiffness considered for torque computation and r is the radius of the probe. So that, if the applied rotation exceeds the following constraint

$$\Delta\theta_{max} = \frac{f_N \mu_s}{k_\tau r},$$

rolling condition is switched to sliding.

4 CONCLUSIONS

This chapter presents a fast computation method for the simulation of soft finger contact model that considers also rolling effects. The algorithm is based on the Euler angle decomposition, used to quantitatively describe the torque exchanged about the normal at the contact and the motion of the contact point due to rolling and the evolution of contact point. The algorithm has been tested and validated with numerical simulations in which the sphere-plane rolling contact was considered. The results obtained with the simplified computation method were compared with those obtained by the integration of the classical rolling formulation, reported in [1]. The numerical results showed a consistent reduction of the computational time, while the precision in the contact point path during the simulation was not significantly affected. The availability of a fast method to simulate rolling effects and soft finger contact models could have different applications. In this paper we presented an example, in which the proposed model was adopted to simulate, by means of an haptic interface, the contact between a virtual fingertip and a plane. The application of the proposed algorithm to haptics is still going on, further validations will be analyzed, the limits of the computational method will be furthermore investigated and correction strategies, needed to avoid the drift of the predicted trajectories, will be assessed.

Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework Program FP7/2007-2013 under the grant agreement 601165 of the project “WEARHAP” and under the grant agreement 270460 of the project “ACTIVE”.

References

1. R. M. Murray, Z. Li and S. S. Sastry, A Mathematical Introduction to Robotic Manipulation, Boca Raton, FL: CRC, 1994.
2. D. Prattichizzo and J.C. Trinkle, Grasping. In: Springer Handbook of Robotics. Springer, 2008.
3. K. L. Johnson, Contact Mechanics, Cambridge University Press, 1987.
4. F. Barbagli, A. Frisoli, K. Salisbury and M. Bergamasco, Simulating human fingers: a soft finger proxy model and algorithm, HAPTICS ’04. Proceedings. 12th International Symposium on, pp. 9-17, 2004.
5. F. Barbagli, D. Prattichizzo, and J. K. Salisbury, A Multirate Approach to Haptic Interaction with Deformable Objects Single and Multipoint Contacts, The International Journal of Robotics Research, vol. 24, no. 9, pp. 703-715, 2005.
6. H. Goldstein, Classical Mechanics (2nd ed.), Reading, MA: Addison-Wesley, 1980.
7. A. Frisoli, F. Barbagli, E. Ruffaldi, K. Salisbury and M. Bergamasco, A Limit-Curve Based Soft Finger god-object Algorithm, Haptic Interfaces for Virtual Environment and Teleoperator Systems, 14th Symposium on, pp. 217-223, 2006
8. Y. Yin, T. Sugimoto and S. Hosoe, MLD Modeling and MPC of Hand Manipulation, IEICE Trans. Fundamentals, vol. E-88-A, no.11, 2005.
9. Z. Li and J. Canny, Motion of two rigid bodies with rolling constraint, Robotics and Automation, IEEE Transactions on, vol.6, no.1, pp.62-72, 1990.
10. On-line: http://www.forcedimension.com.
11. On-line: http://www.chai3d.org.
12. D. C. Ruspini, K. Kolarov, and Khatib, “The haptic display of complex graphical environments.” Computer Graphics, 31(Annual Conference Series):345-352, 1997.