Handling Frictional Contacts on Smooth Surfaces using an Evolving Contact Approach

Luning Fang, Dan Negrut

Department of Mechanical Engineering, University of Wisconsin - Madison, 1513 University Ave, Madison, WI-53705
{lfang9,negrut}@wisc.edu

EXTENDED ABSTRACT

1 Introduction

We report on a modeling technique for handling continuous frictional contacts between smooth surfaces encountered in multibody dynamics. Additional constraints describing the contacts are included in the Differential Algebraic Equations (DAEs). The solution to the DAEs simultaneously produces the normal contact force and local contact points. Frictional loads, including sliding, rolling and spinning friction loads, are evaluated explicitly. The salient feature of this approach is that since contact points are solved as unknowns, collision detection is no longer needed once the contact initiates. Moreover, there is no need for meshes, since the method is able to treat surfaces in contact in close form or piece-wise analytically.

2 Equations of Motion with Contact Constraints

The state of a system of nb rigid bodies, each of mass m_i and inertia matrix $\bar{\boldsymbol{J}}_i$, can be described by translational and angular acceleration, $\ddot{\boldsymbol{r}}_i$ and $\dot{\bar{\boldsymbol{\omega}}}_i$; velocity, $\dot{\boldsymbol{r}}_i$ and $\bar{\boldsymbol{\omega}}_i$; and position $\boldsymbol{r}_i$ and orientation $\boldsymbol{A}_i$, where $i = 1,2,\ldots,nb$. Herein, a letter with a bar $\bar{\bullet}$ represents a quantity expressed in the Local Reference Frame attached to the body. Locally, the surface of body i is described as $g_i(\bar{\boldsymbol{\alpha}}_i) = 0 \in \mathbb{R}$. For a smooth surface, at any point $\bar{\boldsymbol{\alpha}}_i$, there exists a unique outward-pointing normal $\bar{\boldsymbol{n}}_i = \nabla g_i(\bar{\boldsymbol{\alpha}}_i)$. When bodies i and j are in contact at point P, assuming no penetration, $\boldsymbol{r}_i^P$ and $\boldsymbol{r}_j^P$ coincide as seen in Fig. 1a,

$$\boldsymbol{\Psi}_{ij} : \boldsymbol{r}_i^P - \boldsymbol{r}_j^P = \boldsymbol{r}_i + \boldsymbol{A}_i\bar{\boldsymbol{\alpha}}_i - \boldsymbol{r}_j - \boldsymbol{A}_j\bar{\boldsymbol{\alpha}}_j = \boldsymbol{0} \in \mathbb{R}^3. \tag{1}$$

Additionally, the unit surface normals at P on body i and j, $\boldsymbol{n}_i^0$ and $\boldsymbol{n}_i^0$, point in opposite directions, expressed as,

$$\boldsymbol{N}_{ij} : \boldsymbol{c}(\boldsymbol{A}_i\bar{\boldsymbol{n}}_i^0 + \boldsymbol{A}_j\bar{\boldsymbol{n}}_j^0) = \boldsymbol{0} \in \mathbb{R}^2, \text{ where } \boldsymbol{c} = [1,0,0] \text{ and } [0,1,0]. \tag{2}$$

For each contact, we have a total of 7 constraints, $\boldsymbol{\Psi}_{ij} = 0$, $\boldsymbol{N}_{ij} = 0$, $g_i = 0$ and $g_j = 0$. For a collection of bodies of nc contacts, the total virtual work from all the normal contact forces should be zero, provided that the virtual displacement at the contact point satisfies the contact kinematics constraints,

$$\boldsymbol{\Phi}(\boldsymbol{r}_1,\boldsymbol{r}_2,\cdots,\boldsymbol{r}_{nb},\boldsymbol{A}_1,\boldsymbol{A}_2,\cdots,\boldsymbol{A}_{nb},\bar{\boldsymbol{\alpha}}_i^1,\bar{\boldsymbol{\alpha}}_j^1,\cdots,\bar{\boldsymbol{\alpha}}_i^{nc},\bar{\boldsymbol{\alpha}}_j^{nc},t) = \boldsymbol{0}_{7nc}. \tag{3}$$

By applying D'Alembert's principle, one can derive the constrained equations of motion in the matrix form as

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{r}} \\ \dot{\bar{\boldsymbol{\omega}}} \\ \bar{\boldsymbol{\alpha}} \end{bmatrix} + \boldsymbol{\Phi}_{\boldsymbol{q}}^T\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{F}^a \\ \boldsymbol{\tau}^a - \tilde{\bar{\boldsymbol{\omega}}}\boldsymbol{J}\bar{\boldsymbol{\omega}} \\ \boldsymbol{0} \end{bmatrix}, \tag{4a}$$

$$\boldsymbol{\Phi}(\boldsymbol{r},\boldsymbol{A},\bar{\boldsymbol{\alpha}}) = \boldsymbol{0} \tag{4b}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{7nc}$ are the Lagrangian multipliers associated with the contact constraints that produce normal contact forces, $\boldsymbol{\Phi}_{\boldsymbol{q}}$ is the Jacobian of $\boldsymbol{\Phi}$ with respect to $\boldsymbol{r}$, θ, $\bar{\boldsymbol{\alpha}}$. Given the surface geometry $g(\bar{\boldsymbol{\alpha}})$, one can derive $\boldsymbol{\Phi}_{\bar{\boldsymbol{\alpha}}}$ analytically. Friction loads are included in applied forces and torques, $\boldsymbol{F}^a$ and $\boldsymbol{\tau}^a$. Herein, a friction model that tracks the evolution of the contact points on the surfaces is applied to compute the friction [1], taking into account sliding friction force, as well as rolling and spinning friction torques. This friction model has memory and captures the stick-slip condition via micro-displacements. The approach avoids collision detection once the contact starts, since contact points $\bar{\boldsymbol{\alpha}}$ are part of the solution of the DAEs.

3 Numerical Integration Scheme

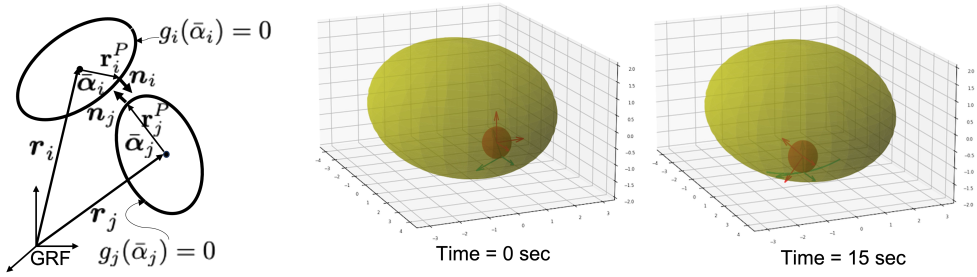
By discretizing Eq.(4) at time, t_n, with a step size of h, we use a first-order half-implicit integration scheme [2] to update the velocity-level quantities, $\dot{\boldsymbol{r}}$ and $\bar{\boldsymbol{\omega}}$, explicitly, and the position-level quantities, $\boldsymbol{r}$ and $\boldsymbol{A}$, implicitly,

$$\dot{\boldsymbol{r}}_{n+1} = \dot{\boldsymbol{r}}_n + h\ddot{\boldsymbol{r}}_n,\ \bar{\boldsymbol{\omega}}_{n+1} = \bar{\boldsymbol{\omega}}_n + h\dot{\bar{\boldsymbol{\omega}}}_n,\ \boldsymbol{r}_{n+1} = \boldsymbol{r}_n + h\dot{\boldsymbol{r}}_{n+1},\ \boldsymbol{A}_{n+1} = \boldsymbol{A}_n \exp(h\tilde{\bar{\boldsymbol{\omega}}}_{n+1}), \tag{5}$$

where $\exp(\cdot)$ is the matrix exponential operation used in Lie-group integrators. By plugging $\ddot{\boldsymbol{r}}_n = (\boldsymbol{r}_{n+1} - \boldsymbol{r}_n - h\dot{\boldsymbol{r}}_n)/h^2$ and $\dot{\bar{\boldsymbol{\omega}}}_n = (\bar{\boldsymbol{\omega}}_{n+1} - \bar{\boldsymbol{\omega}}_n)/h$ into Eq.(4a) at t_n, we derive the following nonlinear system of equations,

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_{n+1} \\ \boldsymbol{\theta}_{n+1} \\ \bar{\boldsymbol{\alpha}}_{n+1} \end{bmatrix} + \boldsymbol{\Phi}_{\boldsymbol{q},n}^T\hat{\boldsymbol{\lambda}}_n = \begin{bmatrix} \boldsymbol{b}_n \\ \boldsymbol{c}_n \\ \boldsymbol{0} \end{bmatrix}, \tag{6a}$$

$$\boldsymbol{\Phi}(\boldsymbol{r}_{n+1},\boldsymbol{A}_{n+1},\bar{\boldsymbol{\alpha}}_{n+1}) = \boldsymbol{0}, \tag{6b}$$



(a) Contact constraints formulated using local contact points, $\bar{\boldsymbol{\alpha}}$.

(b) A sphere placed on a frictional concave surface. The local body frame is colored in red, the contact frame defining contact normal and tangent plane in green, and the contact point trajectory in cyan.

Figure 1: Contact constraints formulation and numerical experiment for smooth surface.

where, $\boldsymbol{b}_n \equiv -(\boldsymbol{M}\boldsymbol{r}_n + h\boldsymbol{M}\dot{\boldsymbol{r}}_n + h^2\boldsymbol{F}_n^a)$ and $\boldsymbol{c}_n \equiv h^2(\tilde{\bar{\boldsymbol{\omega}}}_n\boldsymbol{J}\bar{\boldsymbol{\omega}}_n - \boldsymbol{\tau}_n^a) - h\boldsymbol{J}\bar{\boldsymbol{\omega}}_n$. Newton iterative methods can be used to solve Eq.(6) for unknowns $\boldsymbol{r}_{n+1}$, $\bar{\boldsymbol{\omega}}_{n+1} = \boldsymbol{\theta}_{n+1}/h$, $\bar{\boldsymbol{\alpha}}_{n+1}$ and $\boldsymbol{\lambda}_n = \hat{\boldsymbol{\lambda}}_n/h^2$. The half-implicit scheme does not require the partial derivatives of the friction loads. Additionally, the orientation matrix $\boldsymbol{A}$ is integrated directly and no intermediate generalized coordinates, such as Euler parameters or Euler angles, are needed. Therefore, the Newton iteration matrix of Eq.(6) is easy to assemble.

4 Numerical Experiment and Results

To investigate the evolving contact approach, a sphere of radius 0.5 m and mass 1 kg was set on a concave elliptical frictional surface with an offset from the valley point, see Fig.1b. Initially, the contact point from a sphere-ellipsoid collision can be determined analytically. Two scenarios, with and without rolling friction between the surfaces, were carried out using a step size of 1×10^{-3}. Figure 2 plots the magnitude of the normal contact force $\boldsymbol{F}_N$, the frictional force in the tangential contact plane $\bar{F}_{s,u}$, where $\bar{\boldsymbol{F}}_s = \bar{F}_{s,w}\boldsymbol{w}_0 + \bar{F}_{s,u}\boldsymbol{u}_0$, velocity magnitude, $\dot{\boldsymbol{r}}$, and the Euclidean norm of the contact constraints violation, $\|\Phi\|_2$.

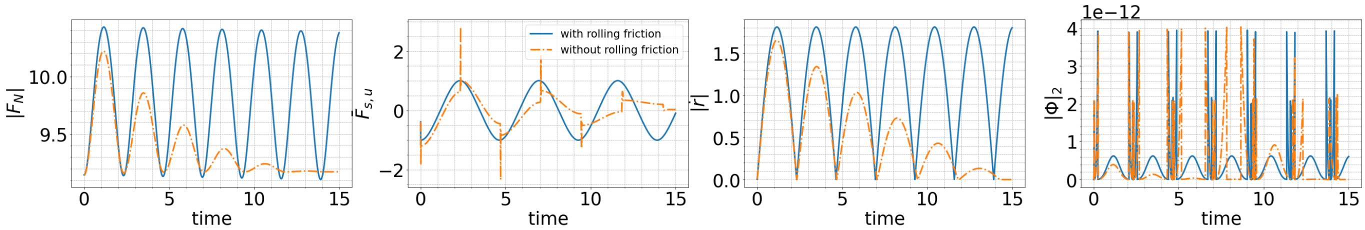


Figure 2: Normal and tangential contact forces, sphere velocity and contact constraint violation over time for sphere rolling on a frictional concave surface, with and without rolling friction taken into account (SI units).

5 Conclusion and Future Work

We propose a novel approach for handling frictional contact among smooth surfaces that splits the treatment of the normal force and the frictional loads. Specifically, extra contact constraints are formulated in DAEs with the Lagrange multipliers producing the normal force. The frictional loads are computed explicitly by tracking the trajectory of the contact points locally. The location of the contact points are solved as unknowns, rather than from the conventional contact detection algorithms, eschewing collision detections after the initiation of the contact. The proposed framework will be implemented in a multi-physics simulation engine Chrono [3] for applications with permanent contacts, such as the cam-follower mechanism or a ball bearing structure.

References

[1] Luning Fang and Dan Negrut. Producing 3D friction loads by tracking the motion of the contact point on bodies in mutual contact. *Computational Particle Mechanics*, 8:905–929, 2021.

[2] Luning Fang, Alexandra Kissel, Ruochun Zhang, and Dan Negrut. On the use of half-Implicit Numerical Integration in Multibody Dynamics. *Journal of Computational and Nonlinear Dynamics*, 18(1), 11 2022. 014501.

[3] A. Tasora, R. Serban, H. Mazhar, A. Pazouki, D. Melanz, J. Fleischmann, M. Taylor, H. Sugiyama, and D. Negrut. Chrono: An open source multi-physics dynamics engine. In T. Kozubek, editor, *High Performance Computing in Science and Engineering – Lecture Notes in Computer Science*, pages 19–49. Springer International Publishing, 2016.