EasyChair Preprint

№ 13313

Collision Detection in the Simulation of Flexible Bodies Using the Floating Frame of Reference Formulation

Xu Dai and Jozsef Kovecses

May 16, 2024

Collision Detection in the Simulation of Flexible Bodies Using the Floating Frame of Reference Formulation

Xu Dai, József Kövecses

Department of Mechanical Engineering
Centre for Intelligent Machines
McGill University
817 Sherbrooke St. West, QC H3A 0C3, Montreal, Canada
xu.dai@mail.mcgill.ca
jozsef.kovecses@mcgill.ca

Introduction

Contact simulation is important for mechanical system models involving interaction between elements. Contact forces can be modelled with constitutive relations or unilateral constraints. In both contact models, the geometric information of the contact, i.e., the contact point position, the direction of contact force, etc., is important to accurately capture the motion of objects during and after contact. Given the configuration and shapes of the objects at each time step of the simulation, certain *collision detection* algorithms are applied to determine whether the contact occurs and to collect the contact information.

Collision detection is more challenging and computationally expensive for mechanical systems with flexible components because the collision boundaries keep changing when the objects are deformed. While the dynamic behaviour of flexible bodies is often modelled with lumped parameters or finite element methods (FEMs), collision detection is a separate geometric problem. Common collision detection algorithms approximate the shape of the objects with multiple elements such as geometric primitives or meshes. Such *element-based* methods have a trade-off between accuracy and efficiency. Specifically, to better represent the collision boundaries of the flexible bodies, more geometric primitives or denser meshes are needed, which usually increases the computational time.

In this work, a *curve-based* collision detection method is proposed, where a group of curves is used to describe the collision boundaries of flexible bodies. When the deformation is not very large, the floating frame of reference formulation (FFRF) is suitable to represent the dynamics and the curve-based collision detection shares the same shape function with the dynamic representation. Preliminary simulation results show that such a curve-based method achieves good accuracy as well as efficiency.

Methodology

The floating frame of reference formulation represents the motion of flexible bodies as the addition of the rigid body motion of the body reference frame and the local deformation [1]. FFRF is suitable for mechanical system models with flexible bodies that have large translation and large rotation, but small deformation. In FFRF, the position of a point B on the flexible body can be written as

$$\boldsymbol{r}_B = \boldsymbol{r}_A + \boldsymbol{R}(\boldsymbol{u}_0 + \boldsymbol{u}_f) \tag{1}$$

where $\boldsymbol{r}_A$ is the position of the origin of the body frame; $\boldsymbol{R}$ is the rotation matrix of the body frame; $\boldsymbol{u}_0$ is the local coordinates of point B in the undeformed state; $\boldsymbol{u}_f$ is local deformation, which is a function of the shape function matrix $\boldsymbol{S}$ and local coordinates $\boldsymbol{q}_f$.

In common element-based collision detection algorithms, the collision boundaries are divided into geometric primitives, which may not be accurate. For example, Fig. 1(a) shows a bending beam whose geometric boundary is approximated by a group of capsules. In this case, the detected contact normal direction $\vec{n}$ can be different from the accurate contact normal $\vec{n}_0$, which may lead to simulation errors.

On the contrary, the curve-based collision detection method describes the geometry of flexible bodies using $\boldsymbol{S}$ in the same way as in the dynamic formulation. In general, at a given simulation time step, the collision detection between two surfaces is essentially checking if the minimum distance between the surfaces is greater than zero. Thus, collision detection can be seen as a minimization problem of a four-parameter distance function between two surfaces, with two spatial parameters for each surface. When

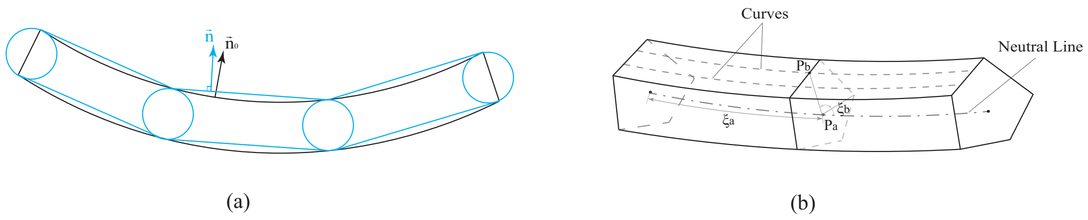

Figure 1: (a) Bending beam approximated by capsules; (b) Curve-based method for beam.

the distance function does not have a close form or is very difficult to formulate, a group of curves on the surfaces can be used to approximate the collision boundaries. Each curve is a function of only one spatial parameter. Thus, the collision detection can be broken down into a set of two-parameter curve-to-curve or curve-to-surface distance function minimization problems. Such a curve-based method can more accurately capture the shape of flexible bodies without significantly increasing computational time. For example, for the beam shown in Fig. 1(b), the position of point P_b on a curve is a function of ξ_a as

$$\boldsymbol{r}_{Pb} = \boldsymbol{r}_A + \boldsymbol{R}(\boldsymbol{u}_{0,Pa} + \boldsymbol{S}\boldsymbol{q}_f + \boldsymbol{R}_{CS}\boldsymbol{u}_{0,PaPb}) \tag{2}$$

where $\boldsymbol{r}_A$ and $\boldsymbol{R}$ is known at the given time step, shape function matrix $\boldsymbol{S}$ and cross-section rotation matrix $\boldsymbol{R}_{CS}$ are functions of ξ_a, local undeformed coordinates $\boldsymbol{u}_{0,Pa}$ and $\boldsymbol{u}_{0,PaPb}$ also only depend on ξ_a.

Case study and results

The simulation of contact between two flexible cylinder beams is used as a case study to compare the collision detection methods. As Fig. 2(a) shows, the blue beam is fixed on one end and the orange beam falls under gravity. Contact between the two beams occurs in the simulation, causing bending and sliding motions. Fig. 2(b) shows the z-coordinate of Point P, where the benchmark solution is created by traversing over the potential contact points on the beam surfaces. In the element-based collision detection method, each beam is approximated by 10 capsules linked together.

Fig. 2(b) shows that the curve-based method achieves more accurate results than the element-based

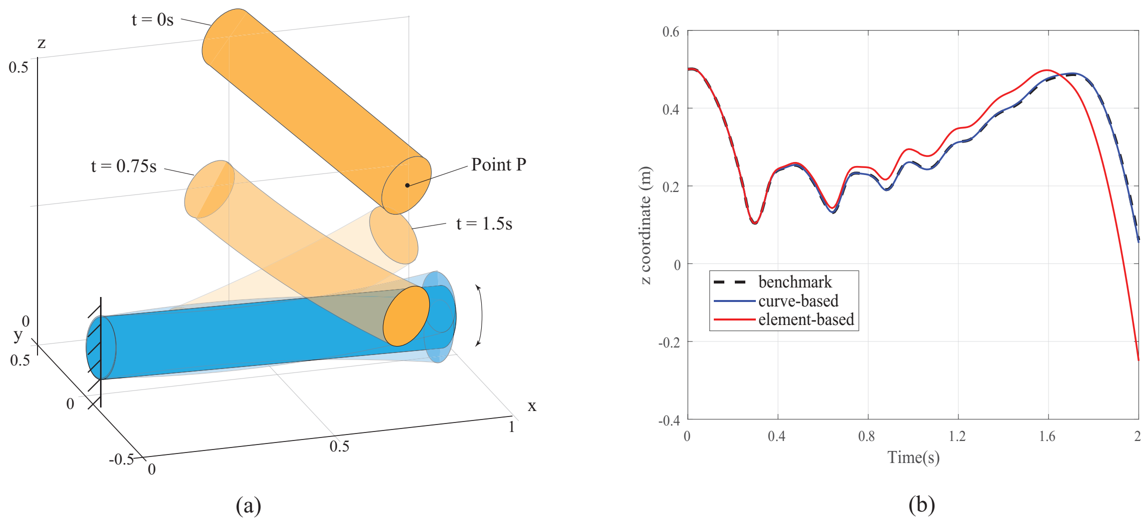

Figure 2: (a) Simulation demonstration; (b) z-coordinate of point P.

method compared to the benchmark. For the 2s simulation, the benchmark takes 7.88s of computational time; the element-based method takes 0.41s; the curve-based method takes 0.35s. The preliminary results show that the curve-based method performs well in both accuracy and efficiency.

References

[1] Shabana, A.A.: Dynamics of Multibody Systems. Cambridge University Press, 2020.