

A Penalty Function-based Modelica Library for Multi-body Contact Collision

Ziheng Zhu¹ Xueqi Ma¹ Dedong Liang¹ XingPei Jiang¹ Yuhui Liu¹ Chen Lu¹ Fanli Zhou¹

¹SuZhou Tongyuan Soft. & Ctrl. Tech. Co. Ltd, Suzhou, China,
{zhuzh, maxueqi, liangdd, jiangxp, liuyh, chenl, zhoufl}@tongyuan.cc

Abstract

Contact collisions are prevalent in mechanical multi-body systems and have always been a significant limiting factor for engineering technology development. This paper examines the fundamental types of contact in multi-body dynamics systems and explores their inherent topological relationships. Based on the multi-body dynamics theory and penalty function contact algorithm, this paper constructed the multi-body dynamics contact model using Modelica, which is a multi-domain unified modeling language. To enhance the applicability of the contact model library in the modeling of multi-body system, the contact model provides a connection interface compatible with the multi-body library in the Modelica standard library.

Keywords: Contact, multi-body dynamics, penalty function, Modelica, MWORKS

1 Introduction

Multibody dynamics primarily investigates the relationships between forces and motion among multiple bodies. In actual multibody dynamics systems, the majority of the relationships between bodies are contact-based. In multibody dynamics simulations, it is challenging to simulate the contact relationships between objects with precision. Consequently, conventional contact relationships are transformed into modeling components, such as motion pairs to enhance modeling efficiency and simulation accuracy. Nevertheless, there exist numerous simulation scenarios where modeling using contact relationships between geometric entities is necessary and cannot be simplified, such as simulation of cam mechanism motion, quadruped robot gait simulation, and Contact process between gears (Dahl M et al. 2017) (Bortoff S A 2020). Currently, basic contact analysis functions are necessary for multibody dynamics software.

Various contact modeling methods have been proposed to address diverse engineering application problems. For instance, Magalhaes H et al. proposed a modeling idea applied for the orbital dynamics contact model based on the Hertzian contact theory (Magalhães H et al. 2020). Additionally, Dahl M et al. introduced a modeling method that applies the Hertzian contact algorithm to the gear contact process (Dahl M et al. 2017). Safaeifar H et al. addressed the issue of hysteresis damping coefficient and proposed a novel modeling approach (Safaeifar H et al.

2020). Bortoff S A et al propose an implicit, event-driven, penalty-based method for modeling rigid body contact and collision in the design and analysis of control algorithms for precision robotics (Bortoff S A 2020).

Referring to the penalty function contact algorithm, this paper constructs a rigid body dynamic contact model library based on the Modelica language and the reusable Modelica mechanical library. Devoted to enhancing the contact analysis capabilities of the Modelica language within the domain of rigid body dynamics. At present, numerous scholars have developed a variety of rigid body dynamic contact model libraries based on the Modelica language. Oestersotebier F et al. employed the geometric analysis method to establish a contact model library specifically for simple contact surfaces (Magalhães H et al. 2020). Furthermore, Hofmann A and Otter M et al. implemented a collision library, utilizing a penalty-based collision approach and incorporates the Bullet Physics Library for collision detection(Hofmann A et al. 2014) (Otter M et al. 2005).

Currently, most mainstream multibody dynamics simulation software features contact analysis function, which is considered as a crucial technical indicator. Multi-body dynamics simulations generally involve the analysis of geometric, mechanical, and mathematical models. The geometric model provides an intuitive representation of the model structure. The mechanical model expands on the geometric model by adding four mechanical elements: kinematic constraints, driving constraints, force elements, and external forces or torques. These elements are assembled according to the kinematic constraints and initial position conditions. This assembly process utilizes a solver to compute the expression and establish a mechanical model. The solver generates the coefficient matrices for the system motion equation of the mechanical model. These matrices can be utilized to derive the system's mathematical model and analyze the model's kinematic and dynamic characteristics.

MWORKS.Sysplorer is a system-level integrated platform for the design and simulation verification of multi-domain industrial products that fully supports the Modelica unified modeling standard across multiple domains. This paper aims to develop a contact model library utilizing the capabilities of this platform.

2 Basic Concept

2.1 Contact Mechanism

Contact friction arises when two distinct surfaces make contact and interact through rubbing, creating a contact state characterized by friction. In the field of physics, surfaces in contact demonstrate the following properties: (1) they prevent penetration; (2) they can transmit both normal pressure and tangential friction forces; (3) they do not transmit normal tensile forces. These characteristics enable free separation among surfaces.

2.2 Penalty Function Method for Contact Mechanism

In the case of contact between rigid bodies, the contact force can be calculated using a spring-damper model that takes into account the penetration depth, resulting in a force, and the penetration velocity, leading to a damping force, between the bodies. To calculate the contact force between bodies in such scenarios, the spring-damper model is frequently employed (Machado M et al. 2012).

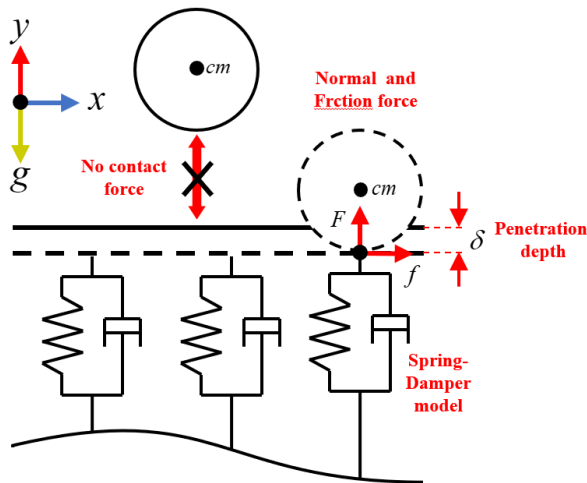


Figure 1 Two-dimensional point-face contact force equivalent schematic

Currently, there are two methods available for computing contact forces in multibody systems: the regression-based and penalty-based contact algorithms. The penalty-based contact algorithm is generally smoother and faster in numerical simulation. The contact model in this study is established solely using the penalty-based contact algorithm. When calculating the contact force between two components using the impact function, it is mainly composed of two parts: the elastic force generated by mutual penetration and the damping force generated by relative velocity. The generalized form can be expressed as:

$$F = K\delta^n + C(\delta)V(t) \tag{1}$$

In which,

F — Normal contact force, measured in N.

K — Contact stiffness, measured in N/m. Generally, a higher stiffness value makes numerical integration more difficult, but if it is too small, the contact situation of the component cannot be accurately simulated. Usually, the contact stiffness is set to 10^8 N/m, The reasonable range of stiffness values can be estimated by Hertz contact theory.

n — Force exponent, used to calculate the contribution value of the material stiffness term in the instantaneous normal force. For contact with high stiffness, $n > 1$ otherwise $n < 1$. For metals, it is usually taken as 1.3~1.5, and for rubber, it is usually taken with a typical value of 1.5.

δ — Penetration depth between objects, a variable that depends on time, measured in mm.

$C(\delta)$ — Contact damping as a cubic function of penetration depth, measured in Ns/m, Generally, 0.1~1% of the stiffness value is taken.

$V(t)$ — Normal relative velocity between contact objects as a function of time, measured in m/s.

In this paper, the contact penetration depth is used to characterize the size of the contact force, mainly considering that the contact model is constructed based on three geometric elements: point, line and surface. It is more intuitive to calculate the contact force between points, lines and surfaces by using penetration depth, and it is easy to expand in building complex geometric contact models.

If the normal penetration depth of the contact point is less than the critical penetration value, the damping force varies cubically with the penetration depth. Conversely, if the penetration depth is greater than or equal to the critical value, the damping force will attain its maximum value, as illustrated in the following figure:

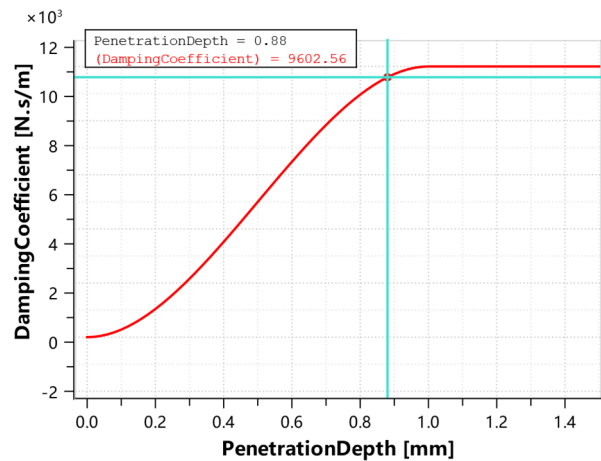


Figure 2 Relationship curve between damping and penetration depth

Firstly, the essence of the penalty function contact algorithm can be analyzed through the point-surface contact model. According to the diagram, contact is deemed to transpire when the distance between the point

and the surface is below zero, signifying the penetration of the two rigid bodies.

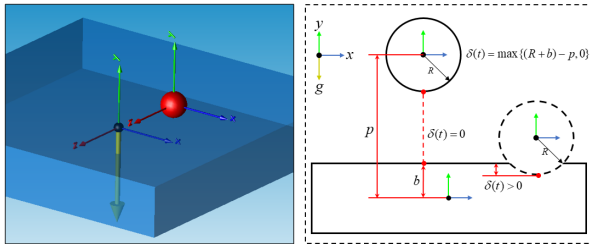


Figure 3 Two-dimensional schematic diagram of point-surface contact

Here, the penetration depth $\delta(t) = \max\{(R+b) - p(t), 0\} > 0$ is used to characterize the degree of penetration between the two rigid bodies. A contact is considered to exist when $\delta(t) > 0$, with larger values indicating a more pronounced penetration phenomenon and resulting in greater interaction forces. In this paper, when the rigid bodies are in contact, the normal pressure generated by the contact between the rigid bodies is equivalently modeled using a spring-damper model, and its value is related to the penetration depth $\delta(t)$ and the relative velocity $V_v(t)$ of the surface normal. The specific expression is as follows:

Contact elasticity:

$$F = \begin{cases} \max\{K\delta^n(t) + C(\delta(t))V_v(t), 0\} & \delta(t) > 0 \\ 0 & \delta(t) \leq 0 \end{cases} \quad (2)$$

Contact damper:

$$C(\delta) = \text{Step}(\delta, 0, 0, \delta_m, C_m) \quad (3)$$

The friction force F_f generated by the contact between rigid bodies is equivalent to a velocity based Coulomb friction model, and its value is calculated according to the contact force F and friction coefficient u . The friction coefficient is related to the relative slip velocity $V_t(t)$ of the two objects in contact (Sextro W et al. 2003).

$$F_f = -F \cdot u(V_t(t)) \quad (4)$$

According to the difference of the relative sliding speed of the two contact objects, the friction process transitions between dynamic friction and static friction. The equivalent formula of friction coefficient and slip velocity is calculated by a cubic step function. The expression of the cubic step function is as follows:

$$\text{Step}(x, x_0, h_0, x_1, h_1) = \begin{cases} h_0 & x \leq x_0 \\ h_0 + \left(\frac{(h_1 - h_0)(x - x_0)^2}{(x_1 - x_0)^2} \right) \left(\frac{3(x_1 - x_0) - 2(x - x_0)}{x_1 - x_0} \right) & (x > x_0 \text{ \& } x < x_1) \\ h_1 & \text{else} \end{cases} \quad (5)$$

The specific expression of friction force is as follows:

$$u = \text{Step}(V_t(t), V_s, -1, V_s, 1) \cdot \text{Step}(\text{abs}(V_t(t)), V_s, C_{st}, V_{tr}, C_{dy}) \quad (6)$$

V_s is the relative slip velocity corresponding to the maximum static friction;

C_{st} is static friction coefficient;

V_{tr} is the relative slip velocity corresponding to the dynamic friction;

C_{dy} is the coefficient of dynamic friction.

The following diagram shows the relationship between the friction coefficient and the relative slip velocity.

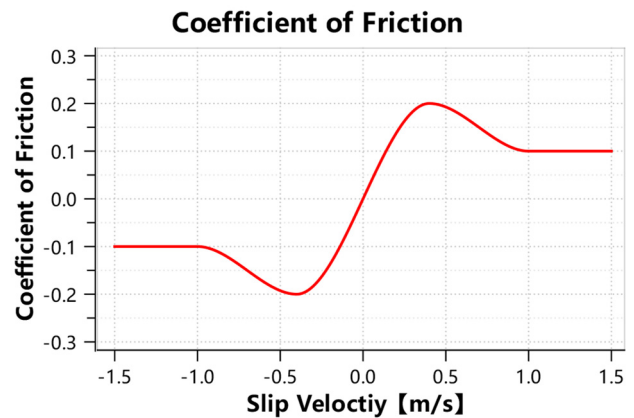


Figure 4 Relation curve of friction coefficient and relative slip velocity

When the slip velocity gradually increases, the friction coefficient gradually increases from 0 to 0.2 of static friction, and then gradually transitions to 0.1 of dynamic friction.

According to the above formula, it can be seen that the value of the interaction force is only related to the velocity vector $V_t(t)$ and the penetration depth $\delta(t)$ between the rigid bodies. The velocity vector $V_t(t)$ can be conveniently obtained using the Modelica multi-body library, and only the penetration depth value $\delta(t)$ needs to be calculated. Therefore, the essence of contact algorithm is actually the process of calculating the penetration depth $\delta(t)$.

3 Overview of Contact Model Library

The contact model library mainly includes examples library, basic component library, friction component library, function library and icon library. The basic component library provides 9 typical contact models, mainly including Sphere-line contact, Sphere-surface contact, and line-line contact. On this basis, six contact models are derived, including line-surface contact, Sphere

-curve contact, Sphere -cylinder, line-cylinder, and Sphere -to- Sphere contact. Each model takes into account the contact force and friction characteristics between rigid bodies, and has a mechanical connection interface, which is compatible with the Modelica multi-body library. The friction component library serves the purpose of calculating the friction force during object contact and forming the contact model components. The function library mainly includes the penalty function contact algorithm function, the damping coefficient calculation function and the geometric analysis algorithm function, etc., which are called repeatedly during the contact simulation. The resulting library structure is illustrated in the accompanying figure.

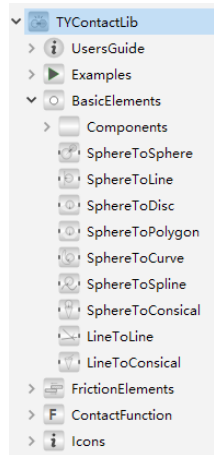


Figure 5 The structure of contact model library

4 Implementation of Contact Models

Based on MWORKS.Sysplorer platform and Modelica mechanical standard library, this paper uses Modelica language to build the contact model mentioned above. The platform has an open Modelica model library customization function to meet different modeling needs. Provides model text, model ICONS, components and other views of the model browse and edit. Support curve display of result data, 3D animation display and rich curve calculation and operation functions.

When analyzing the dynamic process and kinematic process of multi-body contact, the model must describe not only the contact force and contact friction force resulting from the interaction between rigid bodies but also consider the geometric shape information of the objects to determine the contact state and contact area. The following modeling ideas can be employed:

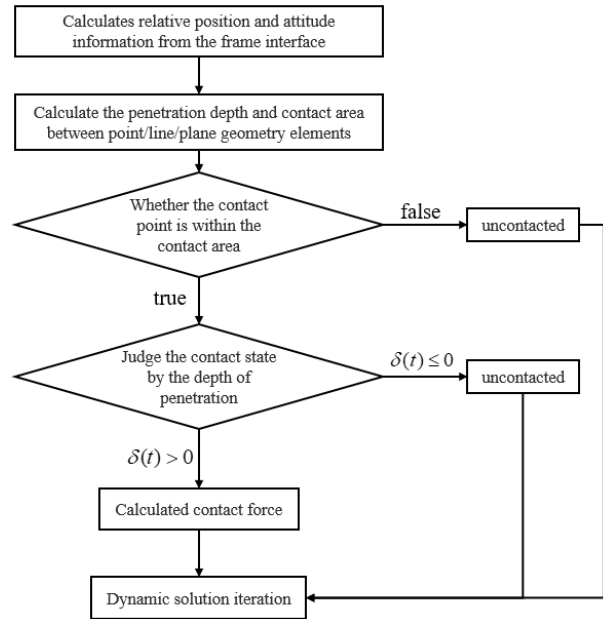


Figure 6 Modeling approach for contact models

The contact model serves as a bridge that connects the multi-body models in the Modelica standard library. It acquires the pose information between rigid bodies through the frame interface, and utilizes the geometric information of the relative local coordinate system between two rigid bodies as parameters to analyze the contact state between their geometric elements.

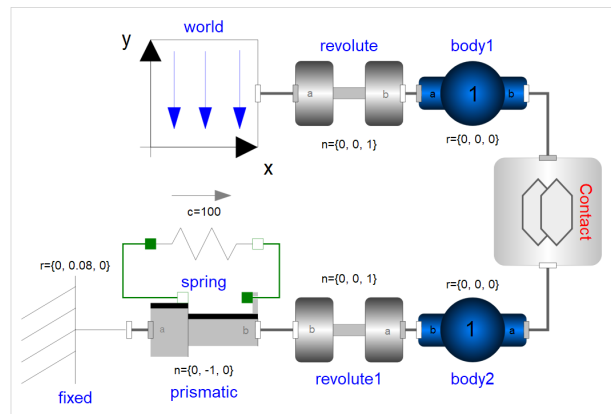


Figure 7 Example of a multibody model including contact

The example model, depicted in the above figure, references the body module from the multi-body library. Users can define the shape of the object according to the actual situation or import geometric models in specific formats. Additionally, users are required to set the geometric data of the object as parameters in the contact model in accordance with specific formats.

5 Contact Detection

The contact algorithm solves only the problem of determining the acting force when contact occurs between rigid bodies. The contact process between rigid bodies involves considering the contact area between the

geometric elements of the rigid bodies. For instance, when dealing with the contact problem between a point and a plane, this article addresses the contact process between a point and a finite surface region. Thus, the key to developing a geometric contact model is determining whether the geometric elements lie within the boundary region.

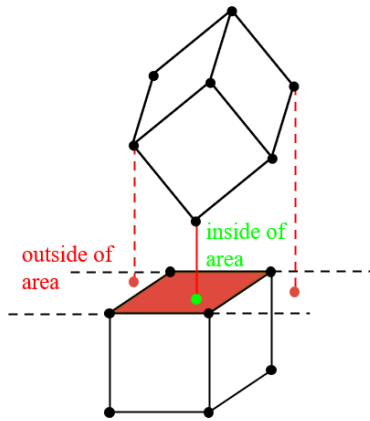


Figure 8 Schematic diagram of point-surface contact area between cubes

This paper mainly refers to the discrete element method and the geometric analysis method to detect the contact area (Oestersötebier F et al. 2014) (Elmqvist H et al. 2015) (Van Den Bergen 2003). Here, this paper based on the theory of geometric graphics, all rigid bodies are described by basic geometric figures such as points, lines, and surfaces, so that the contact problem between rigid bodies can be transformed into the contact problem between three geometric elements. In addition, in order to improve the efficiency of the software to solve the model, all the contact models constructed in this paper use the geometric analysis method to analyze the contact state between objects.

5.1 Point-Line Contact Boundary Detection

Point-line contact involves two main issues: calculating the distance between a point and a line, and determining whether a point is within the line segment. The diagram below shows how to determine if a point lies within the line segment.

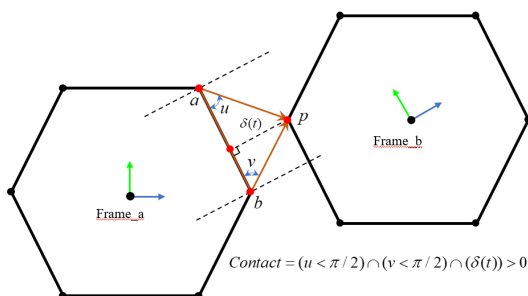


Figure 9 Point-segment region judgment schematic diagram

With frame_a as the reference frame, calculate \overline{ab} , \overline{ap} , \overline{bp} respectively, and then calculate the angle u between the vectors \overline{ab} and \overline{ap} , as well as the angle v between the vectors \overline{ba} and \overline{bp} . When $u < \pi/2$ and $u > \pi/2$, it is judged that the point p is in the accessible area of the line segment ab . When the penetration depth $\delta(t) > 0$ occurs, there will be contact force between the two rigid bodies.

5.2 Point-Surface Contact Boundary Detection

In fact, any polygon or curved surface can be made up of several triangular surfaces. Therefore, solving the problem of determining the region between points and triangular surfaces enables the easy solution of the problem of determining the region of complex geometric bodies. This article uses the centroid method and the schematic diagram below to determine whether the point lies within the region of the triangular surface.

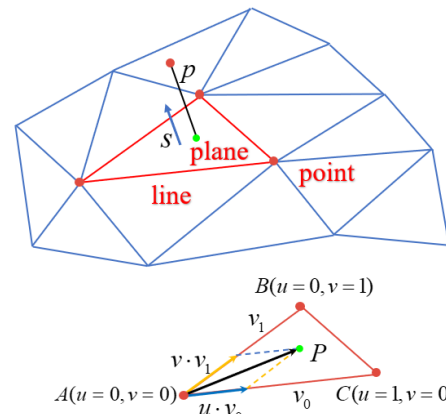


Figure 10 Point and Triangle Face Region Judgment Diagram

Algorithm for determining if a point is within a triangle face region:

$$\begin{cases} u = \frac{(v_1 \cdot v_1) \cdot (v_2 \cdot v_0) - (v_1 \cdot v_0) \cdot (v_2 \cdot v_1)}{(v_0 \cdot v_0) \cdot (v_1 \cdot v_1) - (v_0 \cdot v_1) \cdot (v_1 \cdot v_0)} \geq 0 \\ v = \frac{(v_0 \cdot v_0) \cdot (v_2 \cdot v_1) - (v_0 \cdot v_1) \cdot (v_2 \cdot v_0)}{(v_0 \cdot v_0) \cdot (v_1 \cdot v_1) - (v_0 \cdot v_1) \cdot (v_1 \cdot v_0)} \geq 0 \end{cases} \quad (7)$$

When $u, v \geq 0$ the contact point is inside the triangle area, contact force calculation will be performed. Otherwise, contact force calculation will not be performed. To determine whether a point is inside a rectangular area, it can be divided into two triangular areas, which makes it possible to perform the contact force calculation.

5.3 Line-Line Contact Boundary Detection

The point-surface contact algorithm in the example of contact between cubes only considers the process of vertex-surface contact. To prevent interference issues, as illustrated in the figure, multiple geometric points on the geometric body must be selected for contact calculations with the surface. However, the number of geometric points significantly affects the model's solving efficiency

and accuracy. Therefore, for computational efficiency, an analytical method should be used to establish a line-line contact model instead.

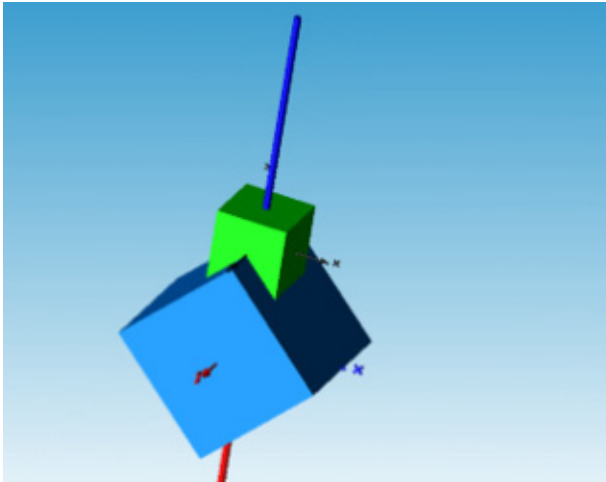


Figure 11 Cubic Contact Interference Phenomenon Considering Only Point-surface Contact

The line-to-line contact model requires consideration of two situations: contact between coplanar line segments and contact between non-coplanar line segments. The former can be achieved by point-to-line contact, while the latter involves determining the positional information of the contact points at each moment, as the contact point between non-coplanar line segments changes over time. To address this, this paper proposes the use of the theory of perpendicular lines, which establishes that there is only one straight line perpendicular to two non-coplanar lines. By calculating the vector information of the perpendicular line, the distance between contact points at each moment and the penetration depth $\delta(t)$ can be determined.

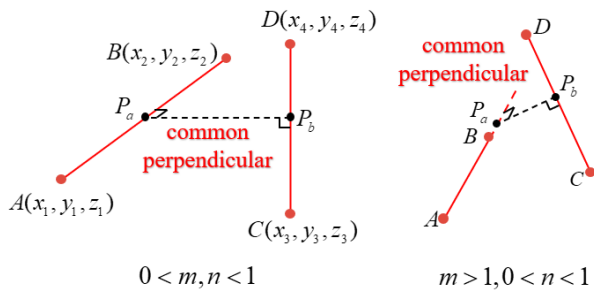


Figure 12 Schematic diagram of line-line contact judgment method

The specific algorithm is shown in the figure above. Assuming that there are two line segments AB and CD in space, P_a and P_b are the intersection points of line segments AB and CD with the common perpendicular, and the coordinate values of P_a and P_b can be described by the following equations.

$$\begin{aligned}
 P_a &\Rightarrow \begin{cases} X = x_1 + m(x_2 - x_1) \\ Y = y_1 + m(y_2 - y_1) \\ Z = z_1 + m(z_2 - z_1) \end{cases} \\
 P_b &\Rightarrow \begin{cases} U = x_3 + n(x_4 - x_3) \\ V = y_3 + n(y_4 - y_3) \\ W = z_3 + n(z_4 - z_3) \end{cases}
 \end{aligned} \tag{8}$$

When $0 \leq m, n \leq 1$, it indicates that P_a and P_b are on line segments AB and CD. Otherwise, P_a and P_b are on the extended lines of the respective line segments AB and CD respectively. With the expressions of P_a and P_b , the distance between the two points can be calculated as follows:

$$P_a P_b = \sqrt{(X-U)^2 + (Y-V)^2 + (Z-W)^2} \tag{9}$$

To find the common perpendicular of two line segments is equivalent to finding the minimum value of P_a, P_b :

$$\min f(m, n) = \sqrt{(X-U)^2 + (Y-V)^2 + (Z-W)^2} \tag{10}$$

Then taking the partial derivative of the function $f(m, n)$ with respect to m, n , a system of two linear equations in two variables can be obtained:

$$\begin{cases} \frac{\partial f(m, n)}{\partial m} = 0 \\ \frac{\partial f(m, n)}{\partial n} = 0 \end{cases} \tag{11}$$

With the above equation system, we can obtain the value of m, n and determine the distance and position vector between the contact points. Utilizing this information, we can apply the contact algorithm formula to develop the line-line contact model. The application of the line-line contact model ensures that the aforementioned interference problem is eliminated in the cubic contact simulation results.

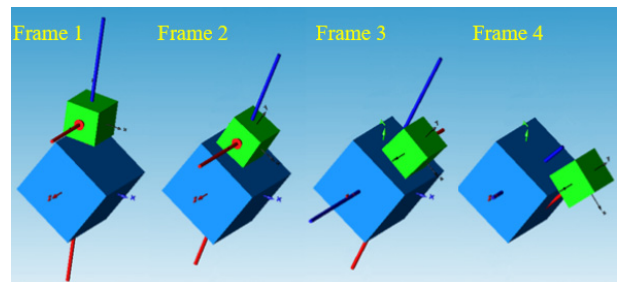


Figure 13 Schematic diagram of point and cylinder contact judgment method

6 Model Extension

6.1 Contact between Point and Cylinder Surface

Drawing on the modeling ideas for point-line contact, it can be extended to the modeling of point-cylinder surface contact, as depicted in the following schematic diagram. By obtaining the position vector of the current point element (P1\|P2) relative to the coordinate system LCI of the cylinder geometry, combined with the position vector of the line element endpoints relative to the cylindrical geometry coordinate system, the spatial geometric information of which line segment the point element is about to contact can be obtained in real-time.

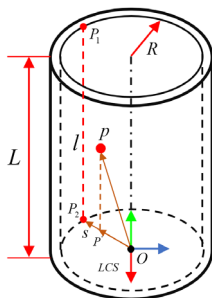


Figure 14 Geometric information diagram of contact line segment

6.2 Point-Curve Contact

The point-curve contact model is equivalent to the point-line contact model with multiple line segment elements, and is based on the point-line contact modeling algorithm. To create the point-curve contact model, geometric information from several points on the curve is read, and cubic spline interpolation is performed to fit the curve. The interpolated curve is then discretized into line segments. By selecting more discrete points, the contact curve can be made to resemble the actual curve more closely.

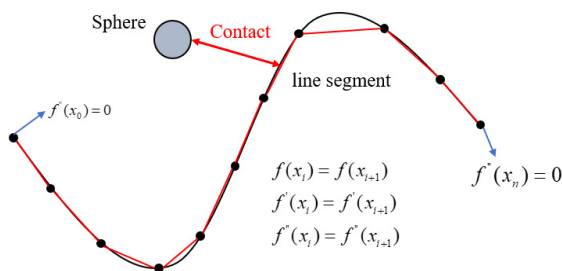


Figure 15 Realization method of contact between point and spline curve

The cubic spline interpolation algorithm actually divides the curve interval $[a, b]$ into n intervals $[(x_0, x_1), (x_1, x_2), \dots, (x_{n-1}, x_n)]$, with a total of $n + 1$ points. Each interval is described by a cubic spline function:

$$f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i \quad (12)$$

The cubic spline interpolation should satisfy the following conditions:

- (1). Each segment should be a cubic spline function;
- (2). It should satisfy the interpolation condition, i.e., $f_i(x_i) = y_i (i = 0, 1, \dots, n)$;
- (3). The curve should be smooth, i.e., $f(x), f'(x), f''(x)$, should be continuous.

7 Simulation Results

The main objective of this chapter is to focus on the aforementioned contact model. It presents three representative examples implemented in MWORKS and utilizes the Dassl algorithm for solving and analysis.

Example 1: Simulating the contact process between a ball with a radius of 30mm and a disk with a radius of 250mm. The z axis of the disk is connected with the world coordinate system by a rotating pair, and a position sine excitation is applied to the rotating pair to analyze the motion of the ball under the action of contact force. The following example demonstrates the construction process.

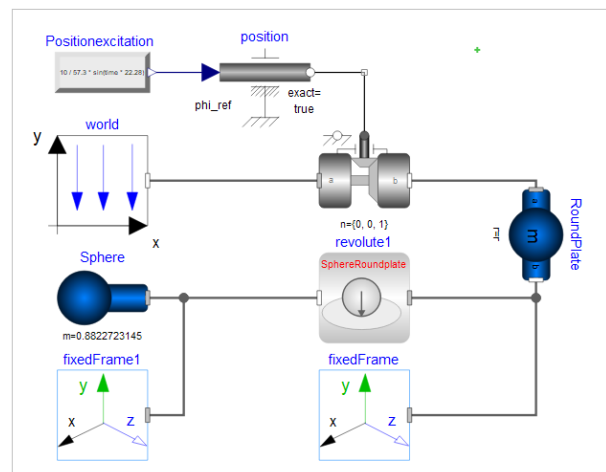


Figure 16 Sphere falling on a swinging round plate

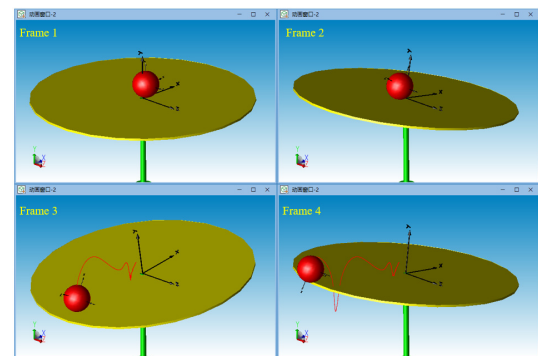


Figure 17 Animation of a sphere falling on a swinging round plate

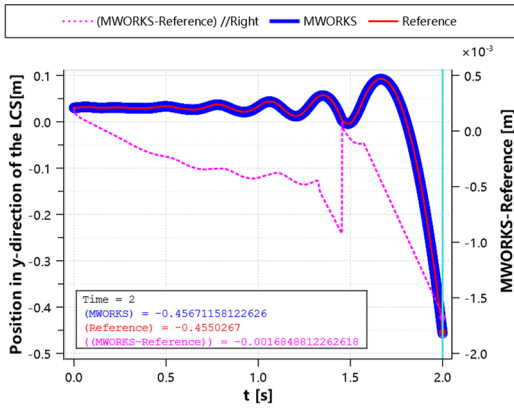


Figure 18 The displacement curve in the Y direction of the center of the sphere

The simulation results are compared with those obtained from Adams, a commercial dynamic simulation software. Figure 17 depicts the simulated curves with 2s. The curves of the two simulation results are in good agreement, and the cumulative error of the position at 2s is 1.6mm.

Example 2: The linear contact model is utilized to simulate the operational principle of the differential. Initially, an equal resistance moment is applied to both wheels, causing them to rotate at the same speed. Subsequently, the resistance moment on the left wheel is increased, leading to the occurrence of differential behavior between the two wheels.

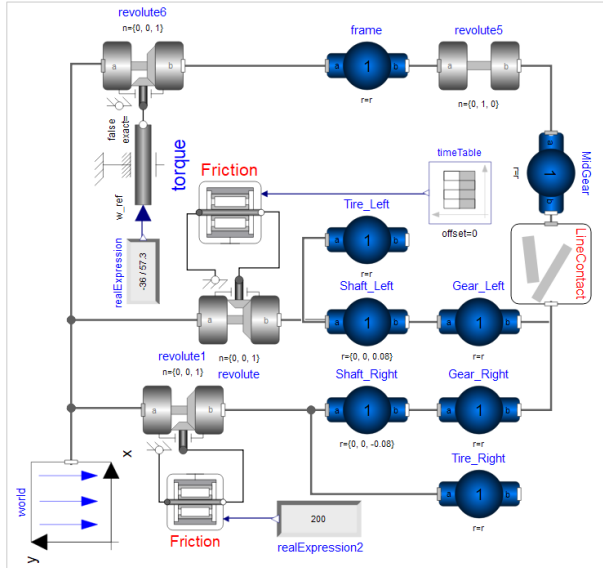


Figure 19 An example of the application of the line contact model

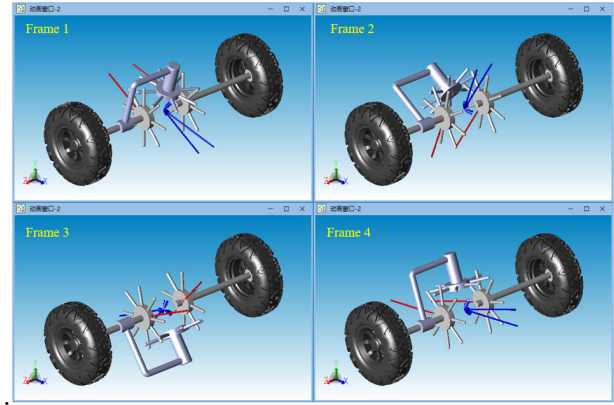


Figure 20 An example animation of a line-line contact model (In the figure, the single arrow is the interacting force vector and the double arrow is the interacting torque vector)

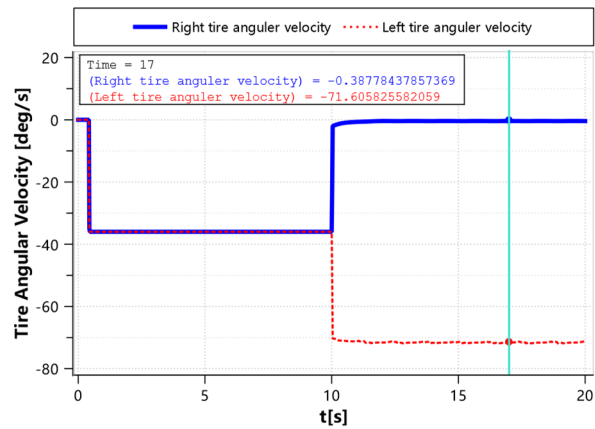


Figure 21 Curve of speed change of two tires

From the graph, it is evident that initially both wheels have the same resistance and speed. However, at 10s, when a resistance moment is applied to the right wheel, a differential phenomenon arises between the two wheels. The speed of the right wheel approaches zero, while the speed of the left wheel doubles its original value.

Example 3: Apply point and curve contact to construct the following two high pair transmission processes.

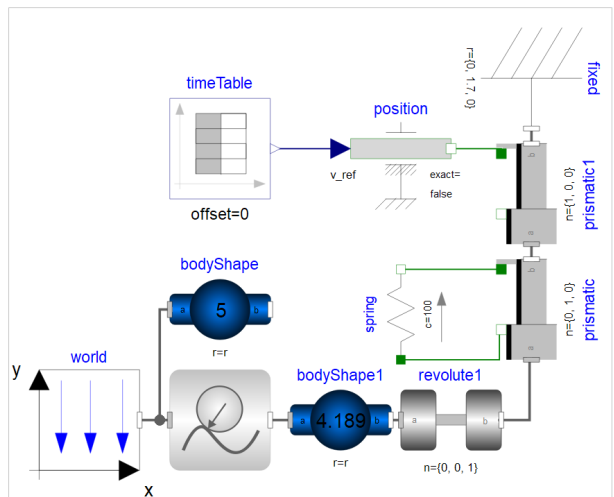


Figure 22 Example of application of point and spline contact model

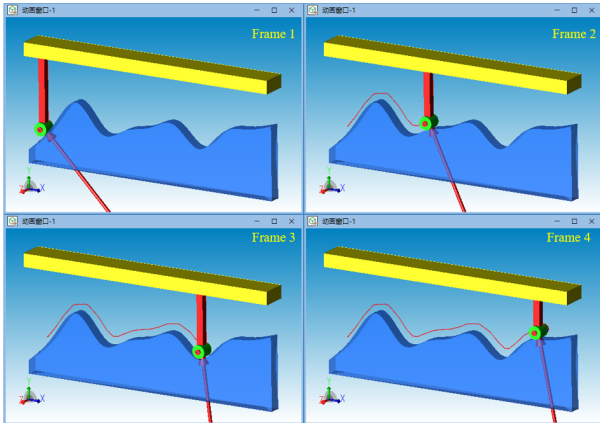


Figure 23 Point and spline curve contact model animation

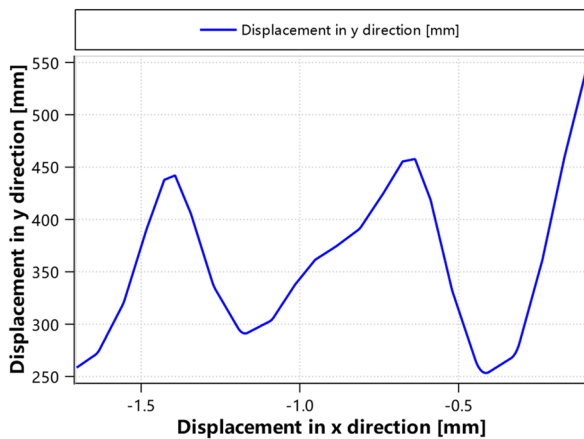


Figure 24 Cylinder displacement curve

Example 4: The point and cylinder contact model is applied to simulate the dynamic contact process of several small balls rolling successively on a funnel until they fall into the cylinder. The following is an example of the build.

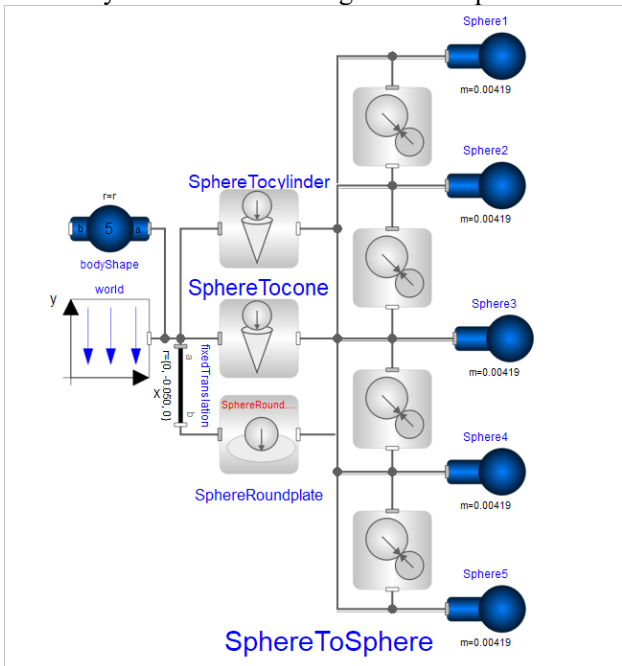


Figure 25 Example of the point-cylinder contact model

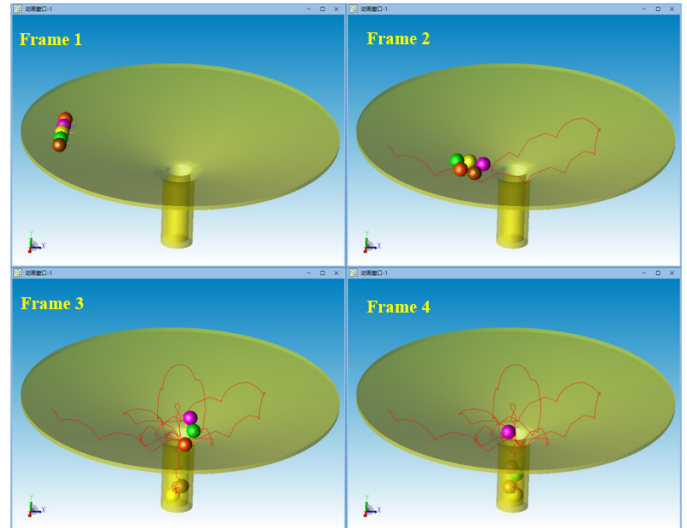


Figure 26 Animation of multiple spheres tumbling down funnels

The example model is constructed by point-cylinder contact model, point-disc contact model and ball-ball contact model. Taking this as an example, multiple typical contact models can be combined to construct more complex contact scenarios.

8 Summary and Outlook

In this paper, the contact library primarily focuses on developing fundamental models for point-surface, point-line, and line-line contacts. These three contact types form the foundation for complex geometric contacts. Furthermore, this paper extends the scope of rigid body dynamic contact models to include more significant variations, such as point contact with spline curves, point contact with cylindrical surfaces, and point contact with polygonal surfaces.

The above discussion represents only a fraction of the broader field of rigid body contact dynamics, leaving significant room for further advancements in contact model development. For instance, when dealing with complex geometric contact problems, it is crucial to optimize model solving efficiency through contact search algorithms, analyze the impact of different contact force calculation methods on the accuracy and efficiency of model solutions, and address contact force errors stemming from the transition between geometric elements.

References

Dahl M , Wettergren H , Tidefelt H (2017). “Modelica Spur Gears with Hertzian Contact Forces”. International Modelica Conference. 2017. DOI: 10.3384/ecp17132755.

Bortoff S A (2020). “Modeling contact and collisions for robotic assembly control”. Proceedings of the American Modelica Conference. 2020.

Magalhães, H. Marques, F., Liu, B. et al (2020). “Implementation of a non-Hertzian contact model for

- railway dynamic application”. *Multibody Syst Dyn* 48, 41–78.
- Safaeifar H, Farshidianfar A (2020). “A new model of the contact force for the collision between two solid bodies”. *Multibody System Dynamics*, 50, 233-257.
- Oestersötebier F, Wang P, Trächtler A(2014). “A Modelica Contact Library for Idealized Simulation of Independently Defined Contact Surfaces”. In *Proceedings 10th International Modelica Conference*, Lund, March (pp. 10-12).
- Hofmann A, Mikelsons L, Gubsch I, et al(2014). “Simulating collisions within the Modelica multibody library”.
- Otter M, Elmqvist H, López J D(2005). “Collision handling for the Modelica multibody library”. In *Proceedings of the 4th International Modelica Conference*.
- Machado M, Moreira P, Flores P, Lankarani H.M (2012): “Compliant contact force models in multibody dynamics: evolution of the Hertz contact theory”. *Mechanism and machine theory*, 53, 99-121.
- Sextro W, Poli C(2003). “Dynamical Contact Problems with Friction: Models, Methods, Experiments and Applications”. *Lecture Notes in Applied Mechanics*, Vol 3[J]. *Appl. Mech. Rev*, 2003, 56(1): B2-B3.
- Elmqvist H, Goteman A, Roxling V, et al(2015). “Generic modelica framework for MultiBody contacts and discrete element method”. In *Proceedings of the 11th International Modelica Conference*, Versailles, France (Vol. 118, pp. 427-440).
- Van Den Bergen, G. (2003). “Collision detection in interactive 3D environments”. CRC Press.