

Article

Variable Time-Step Physics Engine with Continuous Compliance Contact Model for Optimal Robotic Grinding Trajectory Planning [†]

Yongcan Zhou ^{1,2,3} , Yang Pan ^{1,2,3,*}, Junpeng Chen ^{1,2,3} and Tianjian Lei ^{1,2,3} 

- ¹ Shenzhen Key Laboratory of Intelligent Robotics and Flexible Manufacturing Systems, Southern University of Science and Technology, Shenzhen 518055, China; zhouyc2021@mail.sustech.edu.cn (Y.Z.); chenjp2020@mail.sustech.edu.cn (J.C.); 12132267@mail.sustech.edu.cn (T.L.)
- ² Shenzhen Key Laboratory of Biomimetic Robotics and Intelligent Systems, Department of Mechanical and Energy Engineering, Southern University of Science and Technology, Shenzhen 518055, China
- ³ Guangdong Provincial Key Laboratory of Human-Augmentation and Rehabilitation Robotics in Universities, Southern University of Science and Technology, Shenzhen 518055, China
- * Correspondence: pany@sustech.edu.cn
- [†] This paper is an extended version of the published conference paper “Zhou, Y.; Pan, Y.; Lei, T.; Chen, J. Robotic Simulator with High-Precision Perception of Contact Dynamics. In Proceedings of the 2023 IEEE International Conference on Development and Learning (ICDL), Macau, China, 9–11 November 2023; pp. 354–360”.

Abstract: In the transition from virtual environments to real-world applications, the role of physics engines is crucial for accurately emulating and representing systems. To address the prevalent issue of inaccurate simulations, this paper introduces a novel physics engine uniquely designed with a compliant contact model designed for robotic grinding. It features continuous and variable time-step simulations, emphasizing accurate contact force calculations during object collision. Firstly, the engine derives dynamic equations considering spring stiffness, damping coefficients, coefficients of restitution, and external forces. This facilitates the effective determination of dynamic parameters such as contact force, acceleration, velocity, and position throughout penetration processes continuously. Secondly, the approach utilizes effective inertia in developing the contact model, which is designed for multi-jointed robots through pose transformation. The proposed physics engine effectively captures energy conversion in scenarios with convex contact surface shapes through the application of spring dampers during collisions. Finally, the reliability of the contact solver in the simulation was verified through bouncing ball experiments and robotic grinding experiments under different coefficients of restitution. These experiments effectively recorded the continuous variations in parameters, such as contact force, verifying the integral stability of the system. In summary, this article advances physics engine technology beyond current geometrically constrained contact solutions, enhancing the accuracy of simulations and modeling in virtual environments. This is particularly significant in scenarios wherein there are constant changes in the outside world, such as robotic grinding tasks.

Keywords: physics engine; continuous contact model; compliant contact force; robotic grinding trajectory planning



Citation: Zhou, Y.; Pan, Y.; Chen, J.; Lei, T. Variable Time-Step Physics Engine with Continuous Compliance Contact Model for Optimal Robotic Grinding Trajectory Planning. *Sensors* **2024**, *24*, 1415. <https://doi.org/10.3390/s24051415>

Academic Editors: JunZhi Yu, Hesheng Wang and Ning Sun

Received: 28 December 2023

Revised: 16 February 2024

Accepted: 20 February 2024

Published: 22 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the development of robotic trajectory planning technology, a large number of robots [1] have reduced the need for manual labor. The grinding process requires more uncertain contact with the external workpiece, and the physical robot cannot determine whether the planned grinding contact force is appropriate directly. The physics engine [2] is the most important part of the simulator, used to process physical effects such as simulating rigid body motion, collision, gravity, and friction. Simulation needs to be carried out from the virtual environment, but the current physics engine is not accurate enough to pursue

speed. If there are differences between the simulated contact force and the real contact force in the physics engine, the trajectory planning of the sim-to-real [3] process will be wrong.

Contact-impact events encompass intricate physical occurrences characterized by brief durations, substantial forces, rapid energy dissipation, and notable alterations in speed. Modeling these contact-impact phenomena is inherently demanding owing to their significance and complexity. Due to the importance and complexity of these contact-collision phenomena, modeling them is inherently demanding. These events involve numerous variables, such as the material properties of the contacting surfaces, the surface shape of the contacting bodies, and the constitutive relations. The abrupt velocity changes render the contact behavior typical of a nonlinear and non-smooth system.

In the realm of modeling methodology, various approaches have surfaced to describe contact response in the context of multibody dynamics. Broadly categorized, these approaches can be divided into two groups: methods based on “geometric constraints” and models relying on “contact forces” [4]. The “geometric constraint” methods assume that the colliding bodies remain sufficiently rigid to avoid deformation during the collision process. This approach is referred to as the non-smooth dynamic formulation [5]. Conversely, the “contact force”-based model relies on evaluating the spring-damper model as a function of penetration depth and relative velocity, known as compliance or penalty [6]. Unlike non-smooth formulations, the “contact force”-based model is considered compliant, as colliding objects can deform in the contact region. The “contact force”-based model, also known as continuous analysis, continuously evaluates the contact force as a function of penetration.

The linear complementarity problem (LCP) [7] stands as a widely adopted model for addressing impact in the realm of robotics, particularly within the framework of the “geometric constraint” method. In robotic simulations, complementarity is central, requiring either zero displacement or zero contact force. This simultaneous satisfaction implies that the product of two values is forever zero. Essentially, the LCP method is used to solve the geometric contact problem of two objects. The impenetrability of rigid bodies at the velocity level can be achieved by constraining motion or contact forces. Some well-known game physics engines, such as ODE [8], Pybullet [9], and Vortex [10], use equivalent replacements for friction, although inaccuracies arise due to the non-physical nature of the parameters [11]. This kind of replacement for the convenience of calculation is prone to errors in numerical integration, which further leads to the collapse of the system. RaiSim [12] achieved a great breakthrough in computer graphics by implementing a new iterative solution method through the commonly used Gaussian-Sedar (PGS) method. Drake employs its transition-aware solver TAMSİ [13] to accomplish compliant contact with regulated friction. MuJoCo [14] greatly improves the constraint-solving method by relaxing parameters to formalize the problem. Due to its superior computing speed, MuJoCo is widely used in the reinforcement learning training of robots [15].

Adopting a compliant contact force model has proven helpful [16] in overcoming the limitations faced by traditional approaches. The core idea of the continuous compliant contact model is a spring damping model in which the surface is regularly filled with parameters related to the surface material coefficient [17]. According to the set components, the contact force can be achieved to bounce off, thereby preventing penetration at the force level. The determination of stiffness and deflection magnitudes for the spring-damper elements is carried out based on factors such as relative indentation, material properties, and surface geometries of the colliding bodies. After years of development, Flores et al. [18] and Gonthier et al. [19] proposed the continuous compliance contact model. Their model was continuously compared with real physical collisions, and experiments verified the reliability of the model. In the field of mechanical simulation, the method of calculating collision using compliant contact force is widely used due to its speed and ease of programming. At the same time, this model can perform dynamic calculations very well from the physical level, which is beneficial to the numerical stability of the entire system.

Therefore, this model has been applied to many closed-source commercial programs, such as RecurDyn [20], EDEM [21], and ADAMS [22].

Unlike prevalent robotic simulators [23], which often rely on geometric constraints for solving the contact process, thereby confining rigid contact to an idealized state, this paper aims to address these limitations. Our proposed solution involves a variable time-step physics engine with a continuous compliance contact model [24]. Firstly, the bouncing ball experiment with different consumption coefficients was tested to verify the calculation effect of the contact model. Then, by comparing the stability experiments of our proposed method and advanced physics engines, the superiority of the variable time step was verified. Finally, grinding experiments were conducted to further verify the reliability of the conversion from virtual contact force to real. The innovation and primary contributions of this study are outlined below.

- (1) **Development of a Novel Physics Engine:** Central to our innovation is the development of a new physics engine based on the continuous compliance contact model. The analytical solution of the model facilitates the dynamic adaptation of contact forces throughout the simulation, leading to variable time-step integrator behavior.
- (2) **Enhanced Collision Process Modeling:** Our engine introduces a framework to model collision processes precisely. By adopting this advanced approach, we improve simulation-to-real conversion. This is achieved through careful calibration of stiffness and disturbance parameters, which is beneficial to enhancing the simulation accuracy of the entire simulation system.
- (3) **Redefining Conventional Methods:** This study challenges the traditional geometric constraint-based methods, signaling a significant paradigm shift toward more realistic and robust simulations in robotics. By leveraging a swift and stable analytical solution for dynamic equations, our approach ensures rapid acquisition of feedback force.

The rest of this paper is organized as follows. We describe the overview of the whole system in Section 2. We focus on the principle of the contact solver in Section 3. We discuss the performance of ball experiments and robotic grinding experiments in Section 4. Finally, the conclusions of this study are summarized in Section 5.

2. Framework of the Variable Time-Step Physics Engine

In this section, we propose the primary flow chart of the physics engine for robotics (referring to Figure 1). The components of the physics engine include the following: Firstly, Import the CAD model of the multi-jointed robot as a BVH model, which contains vertices and triangles. Then, the collision detection is calculated in FCL and the contact information is obtained, including whether there is a collision, the location of the collision point, and the collision normal vector. Contact forces are calculated in our proposed contact solver. Then, the next link position is further calculated in the integrator used. The dynamical calculation of the next time step is further updated and it continuously calculates the simulation.

Collision detection can provide the fundamental algorithms for contact solvers. Import the configuration of the robot and scene through XML files and save them as triangular patches for broad-phase detection. The data contain the model's vertex and triangle needed for front-end display and collision detection, including vertex coordinates, triangle meshes, texture maps, and information about the object's surface and material. This paper underscores the utilization of bounding volume hierarchies (BVHs), with a specific focus on oriented bounding boxes (OBBs), for the computation of separation distances across various robotic configurations [25]. In collision scenarios involving convex shapes, the Gilbert–Johnson–Keerthi (GJK) algorithm and the expanding polytope algorithm (EPA) stand out as the preferred methods. In continuous collision detection, the goal is to identify potential collisions along a known continuous path. This is achieved by sampling points along the path and performing periodic collision detection at these sampled locations. To enhance the efficiency of these tests, OBBs and rectangular swept spheres (RSSs) are employed as bounding volumes. The practical application of these algorithms is recommended to be carried out using the Flexible Collision Library (FCL) [26].

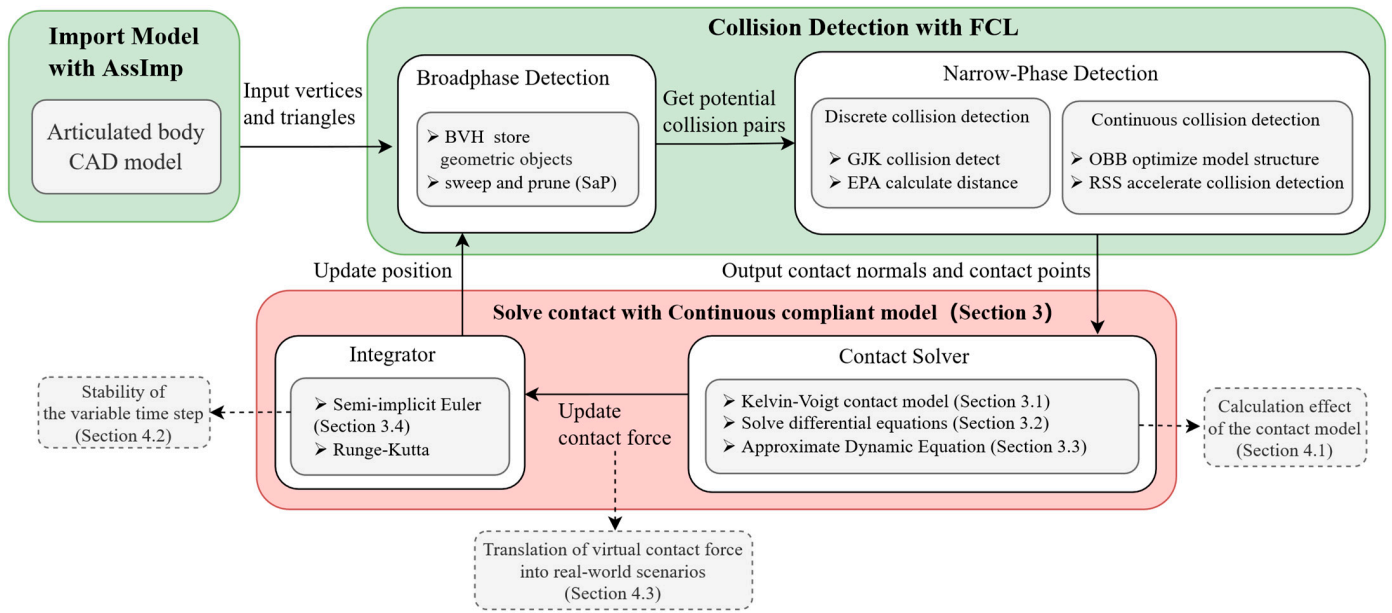


Figure 1. Framework of the variable time-step physics engine.

Utilizing collision detection outcomes, including contact position, normal vector, and penetration depth, this paper proposes a physics engine built upon a continuous compliance contact model. Section 3 is dedicated to detailing the contact solver. Data such as position, velocity, and contact forces obtained from the solver are reintegrated into the dynamical calculations. The robot's position is updated by iterative application of semi-implicit Euler integration, and each new position is updated into the collision detection process.

3. Construction Contact Solver Based on Continuous Compliance Model

3.1. Kelvin–Voigt Contact Model

According to the spring-damping model theory, the contact model can be constructed initially [27]. Taking into account the material properties of the contact surface, the contact position and velocity, and the geometric characteristics of the impact surface, the Kelvin–Voigt model [28] is adopted. Then, the normal force is defined as

$$F_N = D\dot{\delta}(t) + K\delta(t) \quad (1)$$

where $\dot{\delta}(t)$ is the penetration velocity about time t , $\delta(t)$ is the penetration about time t , and D is the viscous damping coefficient. The generalized stiffness parameter K relies on the material properties and the geometry of the contact surfaces. For contact between a sphere i and a plane surface body j , the generalized stiffness parameter depends upon the radius R of the sphere and the material properties of the contacting surfaces, and is expressed as [29]

$$K = \frac{4}{3(\sigma_i + \sigma_j)} \sqrt{R_i} \quad (2)$$

in which the material parameters σ_i and σ_j are given by

$$\sigma_l = \frac{1 - \nu_l^2}{E_l}, \quad (l = i, j)$$

The quantities ν_l and E_l are, respectively, the Poisson's ratio and Young's modulus associated with each object. When it is the initial moment of collision, that is, time t is 0,

we can obtain the penetration $\delta(0) = 0$. From the normal force balance, we can obtain the general dynamic equation as follows:

$$F_{ext} = m\ddot{\delta}(t) + F_N = m\ddot{\delta}(t) + D\dot{\delta}(t) + K\delta(t) \quad (3)$$

where F_{ext} means an external force, such as gravity or other external force acting on the object at the step time. Given the appropriate simulation step size, we can assume that the external force remains constant throughout the time step.

Solving the second-order inhomogeneous equation (Equation (3)) with initial conditions, $\delta(0) = 0$ can obtain the penetration about time t :

$$\delta(t) = e^{rt} \left[\frac{K\dot{\delta}(0) + rF_{ext}}{K\omega} \sin(\omega t) - \frac{F_{ext}}{K} \cos(\omega t) \right] + \frac{F_{ext}}{K} \quad (4)$$

where $r = -\frac{D}{2m}$, $\omega = \frac{\sqrt{4Km - D^2}}{2m}$.

Equation (4) serves as an analytical solution to the dynamical solution, allowing the physics engine to adjust the integration step with a variable time step during the contact process. This adaptive approach significantly enhances the stability of the system compared to geometric constraint solving.

The engine utilizes a combination of local error estimates and stability considerations, which together dictate the adjustment of the time step. The simulation step size is limited by collision detection and other dynamics calculation threads. The thresholds for these adjustments are determined in different scenarios based on empirical studies and theoretical models that predict the simulation's behavior under variable conditions. Therefore, in order to ensure the reliability of the simulation, the simulation step size must be smaller than the minimum time period of other computing threads. By implementing this strategy, we aim to strike a balance between computational efficiency and the fidelity of the simulation outcomes, ensuring that the selected time step remain within bounds that preserve the integrity and reliability of the simulation's dynamics.

3.2. Contact Force Calculation

After obtaining the position (Equation (4)) of the object during the collision process, further calculations are needed to obtain the contact force of the object within the integration step. Integrating the contact force F_N (Equation (1)) within time Δt , we can obtain

$$\begin{aligned} \int_{t_0}^{t_1} F_N dt &= \int_{t_0}^{t_1} (D\dot{\delta}(t) + K\delta(t)) dt \\ &= \frac{(AK + DAr + DB\omega)[\omega \sin(\omega t) + r \cos(\omega t)] e^{rt}}{r^2 + \omega^2} \\ &\quad + \frac{(BK + DBr - DA\omega)[r \sin(\omega t) - \omega \cos(\omega t)] e^{rt}}{r^2 - \omega^2} + F_{ext} t \Big|_{t_0}^{t_1} \end{aligned} \quad (5)$$

where t_0 and t_1 represent the initial and end moments of each contact. And $A = -F_{ext}/K$, $B = (K\dot{\delta}(0) - Kr\delta(0) + rF_{ext})/\omega K$.

In order to find the expression of the contact force model without the viscous damping coefficient ' D ', we need to use energy conservation to further build the contact force model. The simplest way to quantify the energy loss during a contact event is to use the concept of coefficient of restitution, which can be evaluated from the balance of energy that occurs between the initial instant of contact $t^{(-)}$ and the final instant of contact $t^{(+)}$. Equation (3) shows that, since the motion process of spring damping is an analytical solution, the entire collision process time is known as

$$t = \pi \sqrt{\frac{m}{K}} \quad (6)$$

By deriving Equation (4), the analytical solution of velocity can be obtained. Therefore, by incorporating Equation (6) into the analytical solution for velocity, we can obtain the velocity at the final collision moment. And we denote the velocity of the i -th object at the initial collision moment as v_{i0} and at the final collision instant as v_{i1} .

$$v_{i1} = -\dot{\delta}_0 e^{-\xi \frac{\pi \omega}{\omega_d}} \quad (7)$$

where $\xi = \frac{D}{2m\omega}$, $\omega_d = \omega \sqrt{1 - \xi^2}$. According to the principle of energy conservation, the energy change in the collision process is

$$\Delta E = \frac{1}{2} m_i v_{i0}^2 - \frac{1}{2} m_i v_{i1}^2 = \frac{1}{2} m_i v_{i0}^2 \left(1 - e^{-\xi \frac{2\pi \omega}{\omega_d}} \right) \quad (8)$$

Defined by the coefficient of restitution, we have $c_r = -\dot{\delta}^+ / \dot{\delta}^-$. The energy variation within the collision process can be described as

$$\Delta E = \frac{1}{2} m_i v_{i0}^2 - \frac{1}{2} m_i v_{i1}^2 = \frac{1}{2} m_i v_{i0}^2 (1 - c_r^2) \quad (9)$$

Solving Equations (8) and (9) yields the relationship between D and c_r . Subsequently, we can obtain the available collision contact force:

$$F_N = -2|lnc_r| \sqrt{\frac{Km}{\pi^2 + ln^2 c_r}} \dot{\delta}(t) + K\delta(t) \quad (10)$$

By utilizing the analytical solution from the contact model, the proposed physics engine can obtain the compliant contact force throughout the collision. This feature ensures the stability of simulation dynamics calculations and enables the simulator to capture dynamic changes in contact forces accurately. As a result, the simulator offers a realistic modeling of the robotics interactions with its environment, ensuring the reliability of the results.

3.3. Continuous Compliance Contact Model-Based Approximate Dynamic Equation

Hunt and Crossley [30] introduced a comprehensive formulation for contact force, which is expressed as $F_N = K\delta^n(t) + \lambda\delta^m(t)\dot{\delta}^q(t)$. Among them, n , m , and q are all constants. By incorporating a damping term $\lambda\delta^m(t)$ before $\dot{\delta}(t)$, the Kelvin–Voigt model can effectively emulate energy dissipation during the contact–impact process. However, a challenge arises in the Kelvin–Voigt model, as the computed contact force persists even when no deformation occurs in the initial and separation phases of collision, deviating from realistic behavior. Hunt and Crossley’s research has gained widespread recognition and laid the groundwork for the development of numerous other methods for evaluating contact forces. Among them, the most worthy of our attention is novel continuous contact force model [31] with arbitrary indentation depth and velocity exponent using the approximate dynamic equation and system dynamic equation. The coefficient of the damping term $\lambda\delta^m(t)$ can be replaced by a constant. From references [31], we can obtain

$$\begin{cases} m_0 \ddot{\delta}(t) + C_1 K \delta^m(t) = 0 & \text{Compression phase} \\ m_0 \ddot{\delta}(t) + C_2 K \delta^m(t) = 0 & \text{Restitution phase} \end{cases} \quad (11)$$

where the constant term C_1, C_2 is

$$C_1 = \frac{m+1}{n+1} \left(\frac{m_0(n+1)\dot{\delta}^{(-)2}}{2K} \cdot \frac{c_r^2 + c_r^q}{1 + c_r^q} \right)^{\frac{m+1}{n+1}} + \frac{\lambda}{K} \left(\frac{2}{2+q} \right) \left(\dot{\delta}^{(-)} \right)^q$$

$$C_2 = \frac{m+1}{n+1} \left(\frac{m_0(n+1)\delta^{(-)2}}{2K} \cdot \frac{c_r^2 + c_r^q}{1 + c_r^q} \right)^{\frac{m+1}{n+1}} - \frac{\lambda}{K} \left(\frac{2}{2+q} \right) \left(c_r \dot{\delta}^{(-)} \right)^q$$

In order to make the mathematical form easier to understand, the penetration $\delta(t)$ in Equation (11) is replaced by y , and variables of time t are replaced by x . By converting the format, the equation can be converted into the following format:

$$\ddot{y}(x) = Ay^m(x) \quad (12)$$

where $A = -C_i K / m_0$. According to the initial conditions, we have the following constant terms:

$$y(0) = 0, y'(0) = \dot{\delta}^{(-)}$$

The equation is mathematically the Emden–Fowler Equation. The solution can be obtained by looking up the table in [32] as

$$x = \begin{cases} \pm \int (2Ay^{m+1}/(m+1) + C_1)^{-\frac{1}{2}} dy & \text{if } m \neq -1, \\ \pm \int (2A \ln|y| + C_1)^{-\frac{1}{2}} dy & \text{if } m = -1 \end{cases} \quad (13)$$

Equation (13) is the solution to Equation (12), which is also the solution to Equation (11) and obtains the advantages of the Hunt and Crossley contact model. This, in turn, makes the obtained contact force closer to the real force, which further improves the accuracy of the physics engine.

3.4. Semi-Implicit Euler Integrator

In the previous section, we calculated the acceleration of the objects at each moment. In order to further update the physics engine, we need to calculate the velocity and position of the object. The Runge–Kutta integral is orders of magnitude better than the semi-implicit Euler integral for smooth dynamics, but its advantage is lost in the presence of contact dynamics [23]. To speed up the calculation, we employ semi-implicit Euler integration. According to [33], by assuming that the integration step size within Δt is constant, we have

$$\begin{cases} v_{n+1} = v_n + \Delta t \cdot a_n \\ p_{n+1} = p_n + \Delta t \cdot v_{n+1} \end{cases} \quad (14)$$

where the subscript n represents the n th moment calculated by the physics engine, and $n+1$ is the next calculation cycle after one time step corresponding to n , v_{n+1} represents the velocity at moment after an integration time Δt , a_n represents acceleration, v_n represents the velocity at moment n -th. Similarly, p_{n+1} represents the position of the link at the moment after an integration time Δt . It should be noted that, for the purpose of integration accuracy, the position integral in semi-implicit integration is used v_{n+1} .

By incorporating the acceleration into the semi-implicit Euler integration, we can successively update the velocity and position of the robot at each integration step. These updates are then refreshed into the physics engine and synchronized to the display thread. This process enables the achievement of a precise simulation step size. The above is the calculation process of the entire physics engine. The code implementation of the entire calculation process can be expressed as Algorithm 1.

Algorithm 1 Contact Solver base Force

```

Given parameters  $K$  and  $CR$ 
while True do
  Get position of collide object  $\delta_i[t]$ 
  collide(A,B)//Collision detection for object A and B
  if number of contact > 0
    Compute  $I^c, \ddot{r}_c, I^{eff}$ 
    for each contact  $i$  do
      if contact is start
        then ccd  $p_i[t] \leftarrow p_i[t - 1]$ 
      else if contact is end
        then  $t \leftarrow t_c$  //modify time of contact
      end if
      Compute  $a_i[t], v_i[t], p_i[t]$  with integrator
      Compute  $f_i[t] \leftarrow F_i[t]/\Delta t$ 
    end for
  end if
end while

```

4. Experiment and Discussion

In order to verify the reliability of the simulation system with the above contact model, we initially conducted a bouncing ball experiment with varying coefficients of restitution to validate the accuracy of the contact model calculations. Subsequently, we compared the stability of our proposed method with advanced physics engines through stability experiments, confirming the superiority of the variable time-step approach. Finally, grinding experiments were performed to further validate the reliability of translating virtual contact forces into real-world scenarios.

4.1. Bouncing Ball Experiment

Designing bouncing ball experiments to verify contact models is a very effective and conventional method. The classic bouncing ball experiment [28] was selected for its exemplary capacity to illustrate the effectiveness and precision of our contact model in handling dynamic events, a critical focus of our research. At the same time, in the future we will also conduct a comprehensive analysis of the importance of a wider range of experiments to more strongly substantiate our claims.

As shown in Figure 2, a demonstration shows a sphere engaging in a perfect collision with the plane, starting from a height of 1.0 m. This sphere has a mass of 1.0 kg, a radius (R) measuring 0.1 m, and a moment of inertia of $0.1 \text{ kg}\cdot\text{m}^2$. The effective stiffness is $140 \times 10^6 \text{ N/m}$, determined using the formulation in Equation (1). The initiation of the sphere's motion is solely attributed to the gravitational force acting in the downward ($-y$) direction from its starting position. Consequently, the ball undergoes free fall until it encounters the stationary and rigid ground, initiating a collision. Upon contact with the plane, a force is generated, prompting the sphere to rebound. This rebound imparts a specific height influenced by the coefficient of restitution.

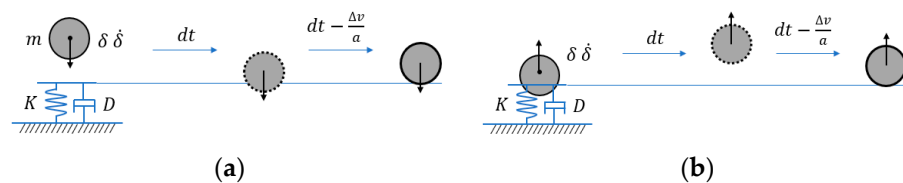


Figure 2. Continuous collision optimization initiation and end process: (a) The initiation moment of the collision; (b) The end moment of the collision.

During the penetration process, the colliding object will not be able to detect the contact surface if the step time is too long. Hence, the penetration process needs to be corrected, as

shown in Figure 2a, using the movement process of the object before the contact collision to find a more accurate contact collision point. Similarly, at the end of penetration, as shown in Figure 2b, the contact collision process can be accurately modeled by correcting the collision time of objects according to the differential equation. This method is called ccd (continuous collision detection).

To investigate the influence of the coefficient of restitution (CR), a series of experiments on the contact dynamic response was conducted, which is depicted in Figure 3a–f. These graphical representations include time-dependent plots illustrating penetration, penetration velocity, and contact force observed during the initial contact. Additionally, Figure 3d illustrates the relationship between contact force and penetration. The analysis incorporates five different CR values: 0.2, 0.4, 0.6, 0.8, and 1.0. It is evident that, as the CR increases, the penetration of the sphere increases, accompanied by a rise in the maximum value of the contact force. This phenomenon results in a prolonged contact duration, and an enlarged hysteresis loop size, representing the energy dissipated during the penetration process. When the coefficient of restitution reaches 1, adhering to pure Hertz contact law, there is an absence of energy dissipation during penetration.

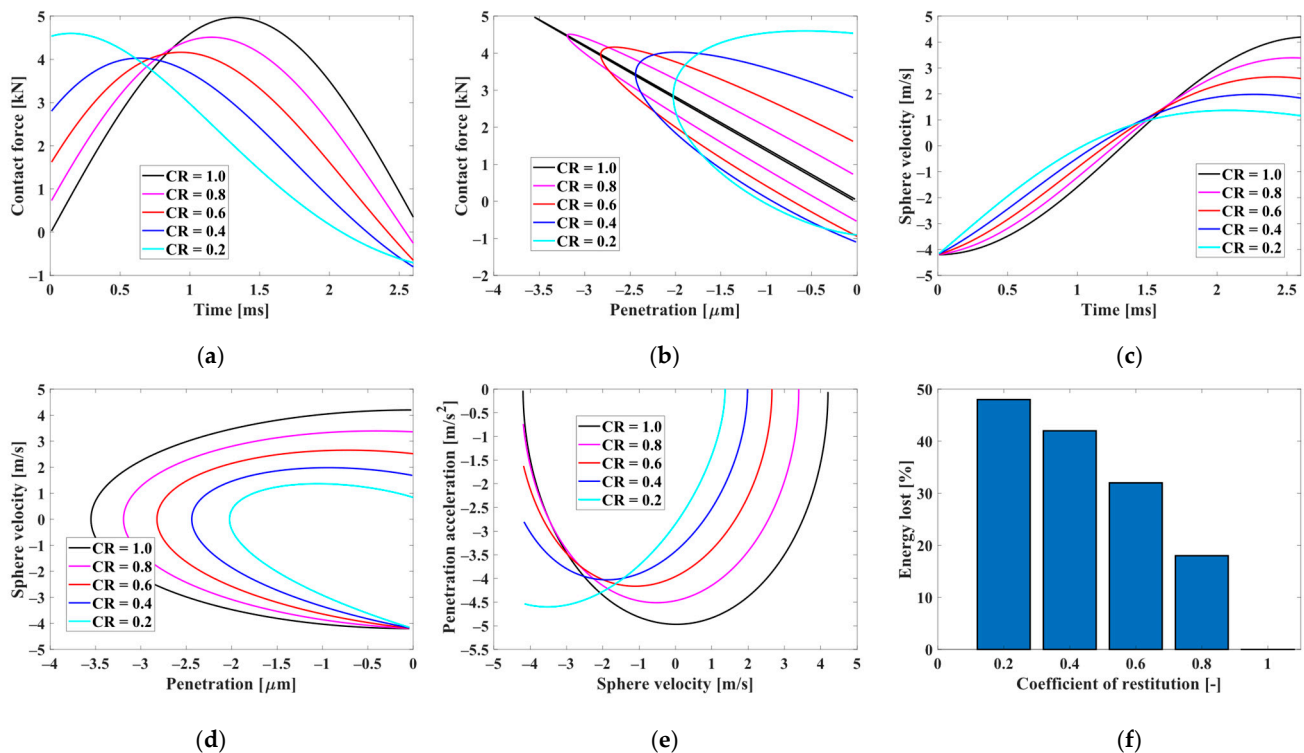


Figure 3. Modeling collisions between balls and surface using the Kelvin–Voigt contact force model at different coefficients of restitution (CR): (a) Contact force plotted against time; (b) Contact force plotted against penetration; (c) Sphere velocity plotted against time; (d) Sphere velocity plotted against penetration; (e) Penetration acceleration plotted against sphere velocity; (f) Energy dissipation percentage.

4.2. Performance of Calculation Stability

In a study conducted by [23], the stability of calculations in different physics engines for robotics was compared. The findings reveal a gradual onset of instability in the calculations as the step time increases. However, the proposed contact force calculation process, employing a continuous compliance contact force model, allows for the variable time step throughout the calculation process, ensuring the stability of the system. In order to detect the maximum simulation step time, we used the bouncing ball experiment when the coefficient of restitution was 1.0. By adjusting the time step, the experimental results can be obtained as shown in Table 1. It shows the maximum limit of each physics engine

in adjusting the simulation step time. We can see that the proposed physics engine can achieve larger step size simulations by using the continuous compliance contact model, and the failure is due to the collision detection speed limit. Due to the advantage of using the contact model, the analytical solution calculated from Equation (4) furthermore ensures that our integrator ensures that the integration time step of the simulation is variable.

Table 1. Comparison between the proposed and other advanced physics engines.

Physics Engine	Contact Solver	Max Timestep (ms)
Bullet [9]	MLCP	1/32
ODE [8]	LCP	1/4
PhysX [34]	LCP/PGS	2
Mujoco [14]	Newton/PGS/CG	16
Proposed	Continuous Compliance Model	128

Observing Figure 4, it becomes evident that, when the coefficient of restitution reaches the unit, the energy within the contact model should be constant. When calculated by integration, a significant energy loss of 12% is observed using the kinetic energy change compared with ODE78 in Matlab. In cases involving multiple collisions, this cumulative loss can result in significant calculation errors, ultimately leading to highly inaccurate results.

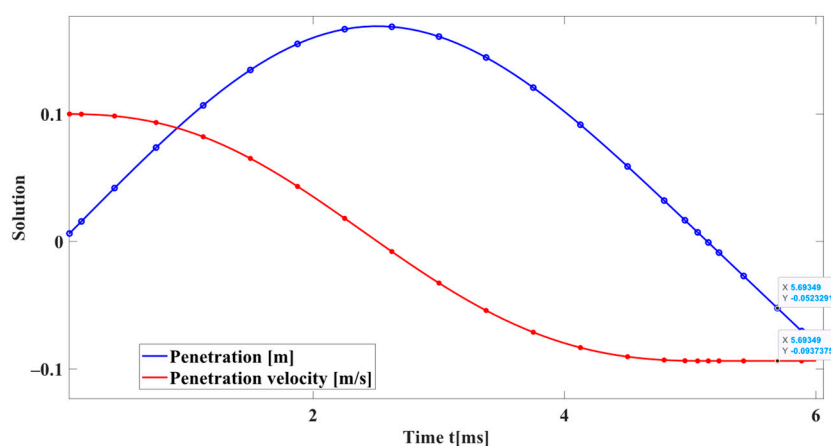


Figure 4. Solving a second-order nonlinear ordinary differential equation using the ODE78 method.

4.3. Optimal Robotic Grinding Trajectory Planning

In production and manufacturing, optimized grinding trajectories can be achieved efficiently through simulation. However, it is necessary to have high virtual-to-reality reliability of contact force in the simulation. A grinding experiment is constructed to confirm the conversion effect of the proposed physics engine. The impact of contact force is then verified through roughness to further verify the reliability of the contact model.

During grinding, the abrasive grains are subjected to the resistance of the deformation of the workpiece material and the friction between the abrasive grains and the workpiece surface, resulting in a grinding force. The grinding force can be decomposed based on the relative position of the workpiece and the grinding tool into a tangential component force, a normal component force, and an axial component force, with the normal component force generally being larger [35]. Furthermore, in the continuous contact model, the friction force can be derived from the normal force by formulas, which will be a focus of our future work. Therefore, normal force is currently selected for research, and other forces will be further studied in the future.

Figure 5 shows the entire grinding experiment process. Initialize and set the distance between the robot and the workpiece for the grinding experiment so that the simulation and physics experiment are consistent. Calculate the contact force of the grinding experiment in

the robot simulation. If the designed trajectory meets the requirements, it will be sent to the actual robot for experiment. Verify that the trajectory is correct by measuring the surface roughness at the grinding location. At the same time, use the force sensor to measure the real contact force. And then compare the measured contact force with the calculated contact force, thereby assessing the effectiveness of contact models in the simulator.

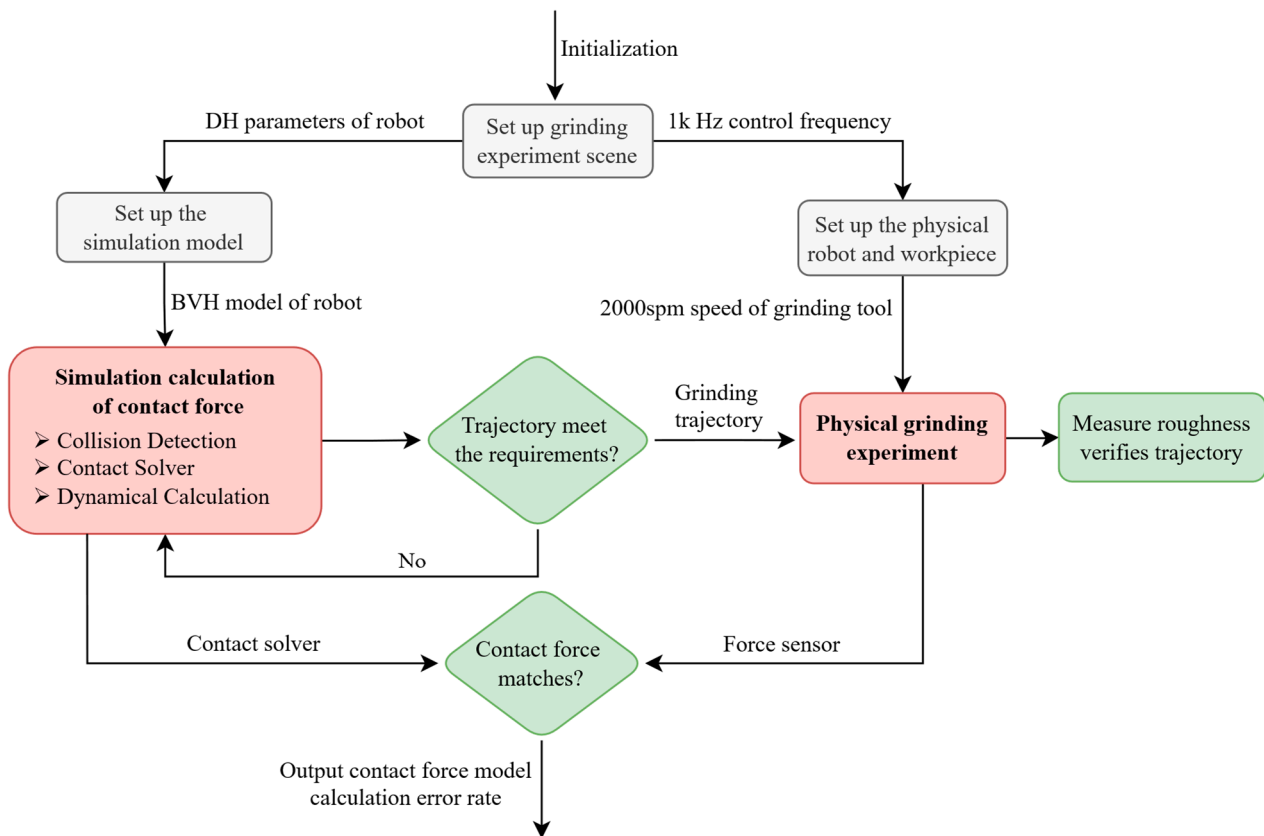


Figure 5. Simulation and physical grinding experimental process.

In this article, we use ESTUN's ER10-900 industrial robotic arm for simulation and physical experiments, as shown in Figure 6.

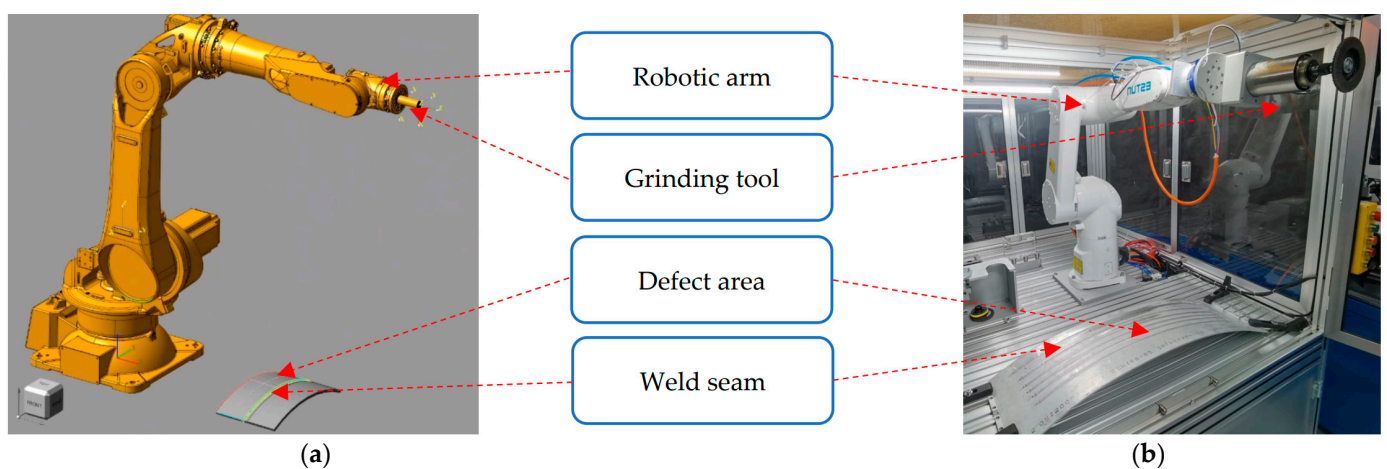


Figure 6. Comparison of experimental platforms: (a) simulation environment; (b) physical environment.

The front-end of the developed robotic simulator uses the open source geometry engine OpenCASCADE [36] as the display. The robotic dynamics algorithm then uses the

open source library ARIS [37] for kinematics and dynamics solving. By starting the display thread, robot dynamics calculation thread, collision detection thread, and contact force solution thread at the same time, the grinding contact force under the determined stiffness coefficient and damping coefficient can be obtained through simulation.

In this article, the method parallel to the first selection curve is used to generate the surface grinding trajectory. The trajectory points are generated based on the surface position and normal vector of the curved surface. The advantage is that it can generate the required grinding trajectory for surfaces with convex shapes, greatly reducing manpower.

Figures 7 and 8 show the simulation and real robot grinding processes, respectively. The trajectory realized in the simulation is sent to the physical robot, and a control frequency of 1 kHz is used in the grinding workstation to ensure real-time performance. This high-frequency control ensures not only precision but also real-time responsiveness during the grinding operation. Due to the utilization of a position control both, a consistent trajectory is observed between the simulated robotic arm and the actual grinding experiment. This synchronization establishes a robust link between the simulated and real-world scenarios, providing a reliable foundation for further analysis and validation of our proposed methods. In the experiment, the choice of 5N was based on preliminary tests designed to study the system's behavior under minimal stress conditions [38]. A force was chosen to be applied in the robotic grinding experiments, recognizing that this number may seem low compared to the higher forces typically used in industrial settings. However, because the grinding experiment can be repeated, the required grinding tasks can be completed in soft metal such as aluminum. Future work will extend the range of forces applied in our experiments to better understand the error behavior and its consequences at these higher force levels.

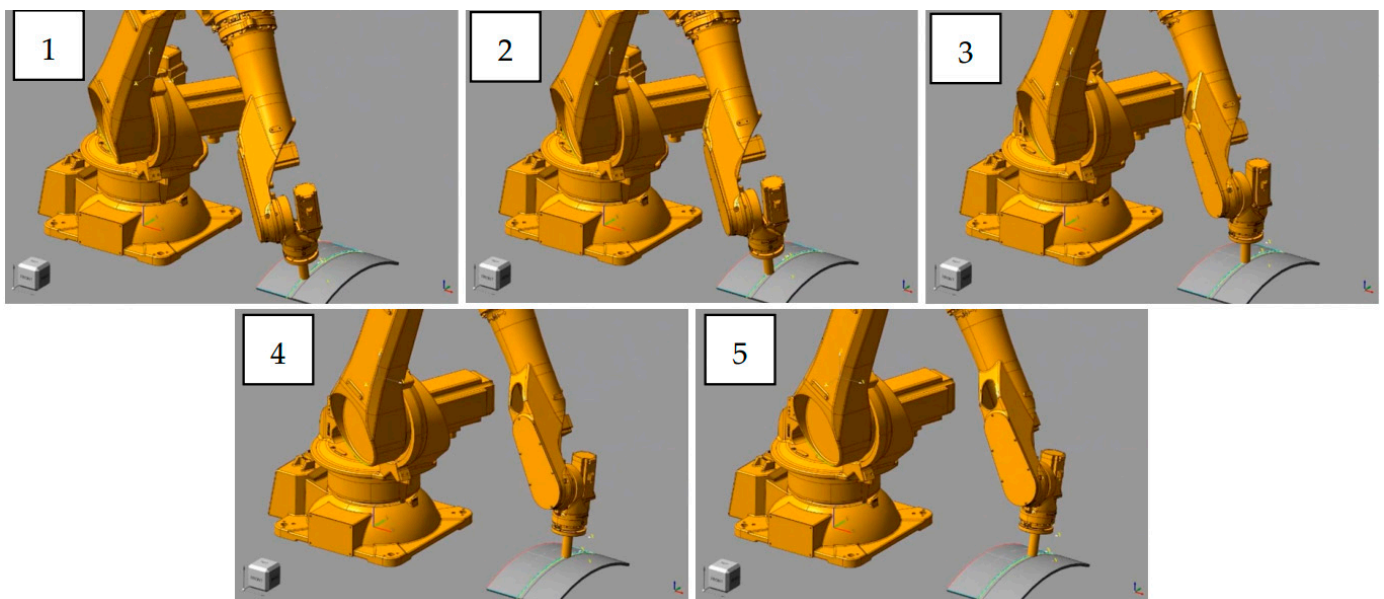


Figure 7. The grinding process by simulation. The numbers 1–5 represent the time sequence of the grinding process.

By installing a force sensor between the end of the robotic arm and the grinding tool, we can measure the grinding contact force in real physical grinding experiments, as shown in Figure 9. Through twenty sets of experiments, we can compare simulated grinding experiments and real physical experiments. Changes in grinding force occur in physical experiments, which is a phenomenon that hardware experiments have to face. This phenomenon is primarily due to the random distribution of the abrasive grains on the grinding tool. These grains are polyhedral in shape, with each edge and corner acting as an individual cutting edge. Real grinding experiments require consideration of many factors, such as grinding workpiece surface, grinding speed, and grinding heat dissipation.

Specifically, in this grinding experiment on soft metal aluminum, the irregularity of the weld and insufficient heat dissipation caused continuous fluctuations in the contact force. Despite these fluctuations, the results confirm that such changes do not affect the accuracy of continuous contact model force calculations in robot simulations, as demonstrated by experimental validation. Obtained through experimental data processing, the difference between the calculated force and the actual contact force was determined by calculating the average relative error, which showed a gap of 2.75%. Pre-calculating contact force in simulations can enhance the efficiency of conventional manufacturing processes to a certain extent.

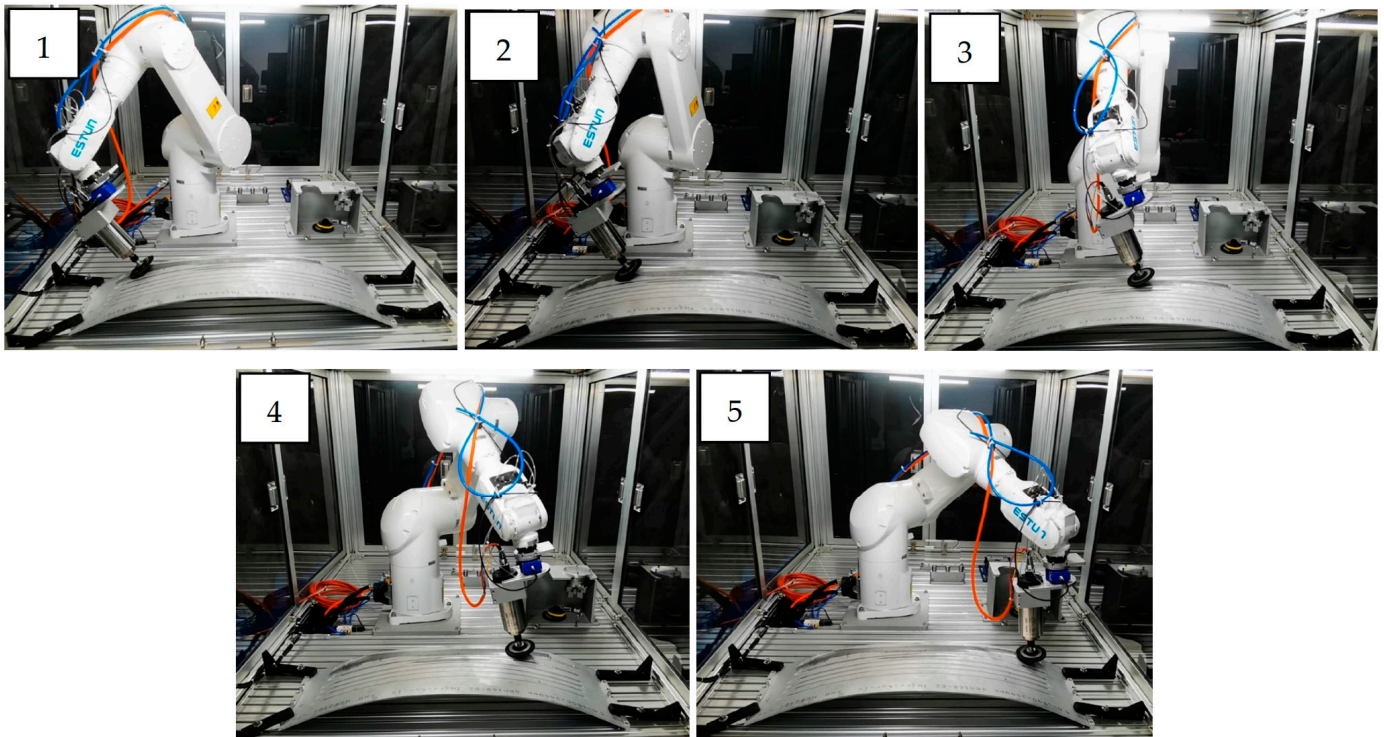


Figure 8. The grinding process by a real robot. The numbers 1–5 represent the time sequence of the grinding process.

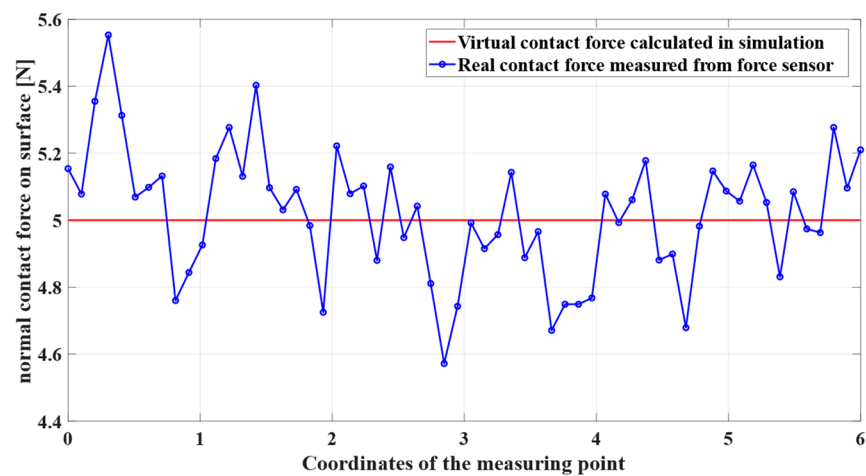


Figure 9. Grinding contact force of the surface.

Recognizing the importance of validating our model, we plan to undertake additional application experiments in the future. These experiments will aim to verify the model's

applicability to more complex, large-scale grinding operations, thereby enhancing its accuracy and reliability in industrial scenarios.

As shown in Figure 10, they are the contact force of two different motion trajectories. It can be seen that the contact force was too small in the first experiment according to the grinding experience database. By adjusting the grinding strategy, the contact force during the entire grinding process can be more reliable with expectations and achieve better grinding results. Future efforts will focus on improving the contact model and physical conditioning. In addition, Section 4.2 demonstrates that variable time steps can be used for relatively fast and stable simulations. Therefore, precomputing and obtaining external forces from the simulation throughout the process facilitates grinding trajectory adjustment and helps improve efficiency in manufacturing.

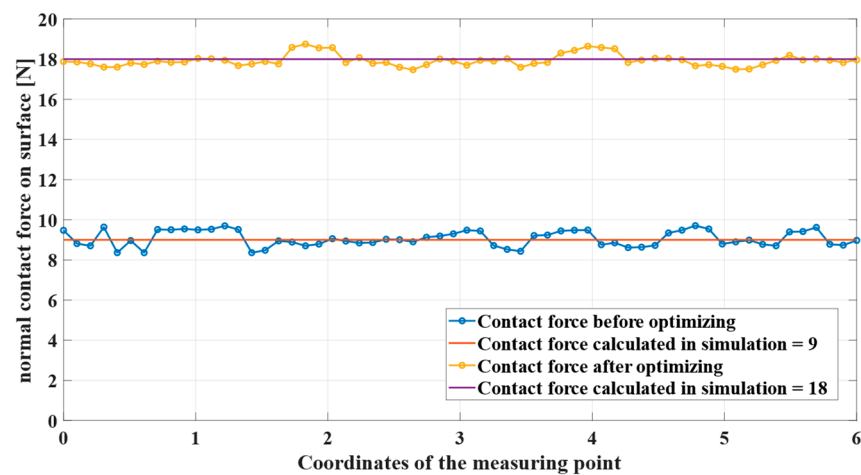


Figure 10. Grinding contact force of the surface before and after optimization.

In order to observe the grinding effect from the side, we detect 10 sets of roughness on the grinding results. As shown in Figure 11, we measure and compare the roughness data of the third point before and after grinding. A total of 10 sets of roughness data were collected from the workpiece surface, revealing compliance with the grinding requirements. Figure 12 presents the roughness data for each point both before and after grinding. Notably, the initial roughness at each point appears to be relatively large, averaging 1.6396 [39]. Following the grinding process, a significant reduction in roughness at each point is observed, with minimal changes, averaging 0.9166. This corresponds to a 44.09% decrease in the average roughness value, affirming the effectiveness of the simulation's strategy.

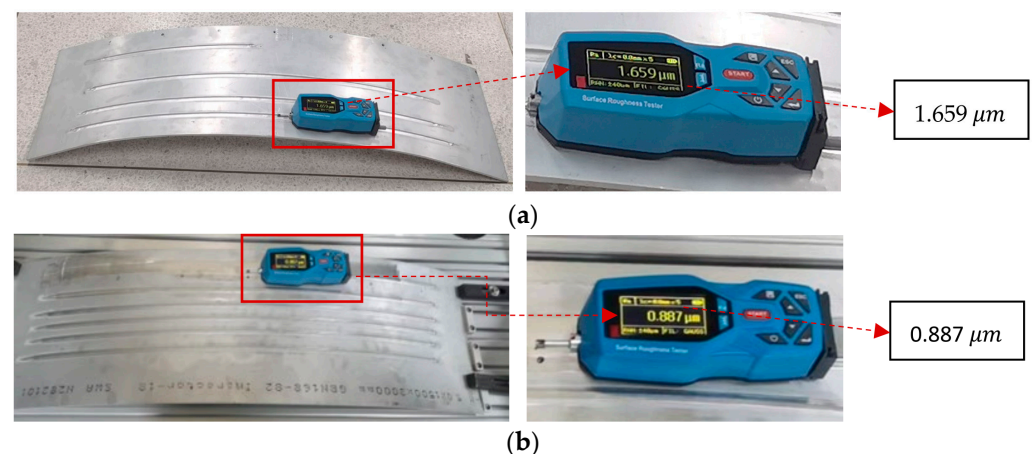


Figure 11. Grinding roughness measurement on the third same point: (a) Surface roughness before grinding; (b) Surface roughness after grinding.

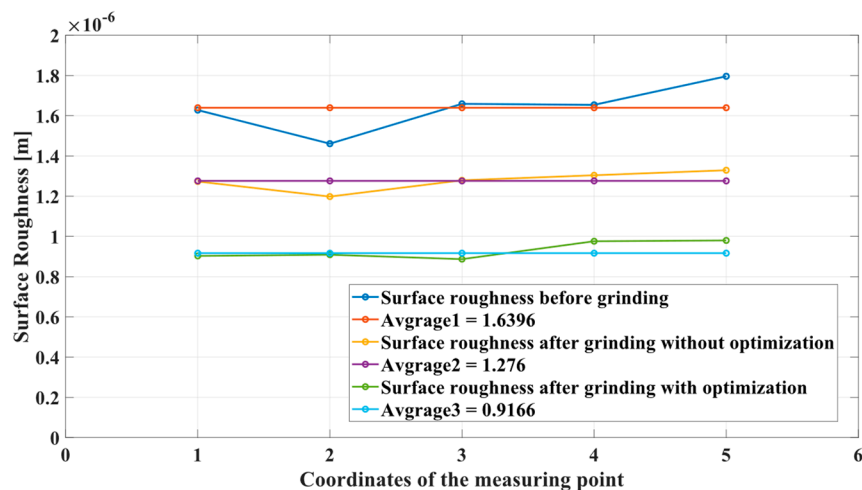


Figure 12. Roughness comparison of each point before and after grinding.

4.4. Discussion

The above experiments verified different aspects of the physics engine and the proposed features.

The bouncing ball experiment (Section 4.1) tested the contact model calculation effect through different coefficients of restitution. When the coefficient of restitution becomes one, it will become a completely elastic collision. The calculation results show that the contact force of the small ball can be accurately calculated as an analytical solution at different penetration depths or different moments.

Comparative experiments (Section 4.2) with different advanced physics engines verify the stability of the variable time step of the physics engine. Comparison of the previous literature [15] found that the maximum time step is only 16 milliseconds, which Mujoco can achieve. By employing a compliance contact force model and solving a second-order differential equation, the simulator achieves 128 ms as a max time step. Unlike the majority of existing robotic simulation engines that depend on geometric constraints to solve the contact process, resulting in hard contacts only existing in an ideal state, our approach employs the continuous compliance contact model with parallel springs and dampers, providing a stable representation of penetration.

We conducted grinding experiments (Section 4.3) on curved surfaces, discovering that optimal roughness during grinding [39] can be achieved by adjusting the contact force. The experimental findings reveal an average relative error of 2.75%, indicating a discrepancy between the virtual force and the actual contact force. In the future, as the contact model becomes more general, such as adding multiple points of contact, we will test experiments in more robotic scenarios. The adjustment not only enhances grinding simulation accuracy but also simplifies the verification process for grinding experiments.

Our results show that the physics engine equipped with the continuous compliance contact model can enable efficient simulations for robotics and in the future also help to effectively generate data for training artificial intelligence algorithms.

5. Conclusions

In this research, we propose a physics engine for robotic grinding, employing a continuous compliance contact model to calculate contact forces during virtual simulations. The contact force solver discussed in this paper relies entirely on the principles of spring damping, providing an analytical solution for contact forces in collision scenarios. This study features an illustration using the classic ball bouncing model, showing the simulator's ability to calculate contact forces continuously. In contrast, contact solutions in game engines, relying on geometric constraints, often involve numerous non-physical parameters, leading to reduced accuracy and stability.

The proposed grinding simulator boasts several key advantages. Firstly, the contact model is grounded in the well-established classic spring–damping model in the mechanical field, incorporating penetration parameters for modeling, resulting in exceptional accuracy. The second advantage is that the simulator is able to obtain the contact forces in the analytical solution, allowing for a large degree of adjustment of the integration time step, thus ensuring more stable system performance. Additionally, analytical solutions to dynamic equations offer significant advantages in terms of computational efficiency and uniqueness. The availability of real-time feedback forces significantly enhances the overall stability of the system, further bolstering its robustness and reliability. Such advantages enable a more accurate simulation of complex grinding processes, aiding technicians in reducing manpower and material resources and ultimately improving experimental efficiency.

One constraint within the scope of this investigation pertains to the measurement of restitution coefficients, especially concerning diverse material surface properties. Consequently, there exists uncertainty in the generalizability of our findings to specific practical scenarios. Addressing this limitation necessitates additional research efforts that concentrate on constructing a comprehensive and dependable correlation model between the spring stiffness system and the damping coefficient. Another viable approach involves the exploration of swifter and more precise methods for identifying the coefficient of restitution, as such advancements enhance the accuracy and reliability of our findings in practical applications. Future work will continue to further optimize model storage and calculation, which will further ensure the speed of collision detection.

Author Contributions: Conceptualization, Y.Z. and Y.P.; methodology, Y.Z. and Y.P.; software, Y.Z. and T.L.; validation, Y.Z., T.L. and Y.P.; formal analysis, J.C.; investigation, Y.Z. and T.L.; resources, Y.P.; data curation, J.C.; writing—original draft preparation, Y.Z. and T.L.; writing—review and editing, J.C. and Y.P.; visualization, Y.Z. and T.L.; supervision, Y.P.; project administration, Y.P.; funding acquisition, Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge the support from the National Key R&D Program of China under Grant 2022YFB4701200, the Science, Technology and Innovation Commission of Shenzhen Municipality under Grant SGDX20220530110804030, JCYJ20220818100417038 and ZDSYS20200811143601004.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article. If you need, you can find the source code at <https://github.com/YongcanZhou/collision> (accessed on 28 December 2023).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chen, J.; Pan, Y.; Yu, J.; Feng, H.; Dai, J.S. Velocity Index and Wading Height Based Design Method of Trajectory Parameters for a Coupled Parallelogram Legged Walking Robot. *Mech. Mach. Theory* **2024**, *191*, 105464. [[CrossRef](#)]
2. Degraeve, J.; Hermans, M.; Dambre, J.; Wyffels, F. A Differentiable Physics Engine for Deep Learning in Robotics. *Front. Neurobotics* **2019**, *13*, 6. [[CrossRef](#)]
3. Siekmann, J.; Green, K.; Warila, J.; Fern, A.; Hurst, J. Blind Bipedal Stair Traversal via Sim-to-Real Reinforcement Learning. *arXiv* **2021**, arXiv:2105.08328.
4. Machado, M.; Moreira, P.; Flores, P.; Lankarani, H.M. Compliant Contact Force Models in Multibody Dynamics: Evolution of the Hertz Contact Theory. *Mech. Mach. Theory* **2012**, *53*, 99–121. [[CrossRef](#)]
5. Leine, R.; Acary, V.; Brüls, O. (Eds.) *Advanced Topics in Nonsmooth Dynamics: Transactions of the European Network for Nonsmooth Dynamics*; Springer International Publishing: Cham, Switzerland, 2018; ISBN 978-3-319-75971-5.
6. Mazhar, H.; Heyn, T.; Negrut, D.; Tasora, A. Using Nesterov’s Method to Accelerate Multibody Dynamics with Friction and Contact. *ACM Trans. Graph.* **2015**, *34*, 1–14. [[CrossRef](#)]
7. Heyn, T.; Anitescu, M.; Tasora, A.; Negrut, D. Using Krylov Subspace and Spectral Methods for Solving Complementarity Problems in Many-Body Contact Dynamics Simulation. *Int. J. Numer. Meth. Eng.* **2013**, *95*, 541–561. [[CrossRef](#)]
8. Smith, R. Open Dynamics Engine. 2007. Available online: <https://www.ode.org/> (accessed on 1 December 2023).

9. Coumans, E.; Bai, Y. Pybullet, a Python Module for Physics Simulation for Games, Robotics and Machine Learning. Available online: <https://pybullet.org/> (accessed on 1 December 2023).
10. CM Labs Simulations Theory Guide: Vortex Software's Multibody Dynamics Engine. Available online: <https://www.cm-labs.com/en/simulation-solutions/vortex-training-simulators/hardware-features/> (accessed on 1 December 2023).
11. Li, J.; Daviet, G.; Narain, R.; Bertails-Descoubes, F.; Overby, M.; Brown, G.E.; Boissieux, L. An Implicit Frictional Contact Solver for Adaptive Cloth Simulation. *ACM Trans. Graph.* **2018**, *37*, 1–15. [CrossRef]
12. Hwangbo, J.; Lee, J.; Hutter, M. Per-Contact Iteration Method for Solving Contact Dynamics. *IEEE Robot. Autom. Lett.* **2018**, *3*, 895–902. [CrossRef]
13. Castro, A.M.; Qu, A.; Kuppuswamy, N.; Alspach, A.; Sherman, M. A Transition-Aware Method for the Simulation of Compliant Contact With Regularized Friction. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1859–1866. [CrossRef]
14. Todorov, E.; Erez, T.; Tassa, Y. MuJoCo: A Physics Engine for Model-Based Control. In Proceedings of the 2012 IEEE/RISJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 5026–5033.
15. Kang, D.; Hwangho, J. SimBenchmark-Physics Engine Benchmark for Robotics Applications: RaiSim vs. Bullet vs. ODE vs. MuJoCo vs. DartSim. Available online: <https://github.com/leggedrobotics/SimBenchmark> (accessed on 1 December 2023).
16. Lidec, Q.L.; Jallet, W.; Montaut, L.; Laptev, I.; Schmid, C.; Carpentier, J. Contact Models in Robotics: A Comparative Analysis. *arXiv* **2023**, arXiv:2304.06372.
17. Anitescu, M. Optimization-Based Simulation of Nonsmooth Rigid Multibody Dynamics. *Math. Program.* **2006**, *105*, 113–143. [CrossRef]
18. Flores, P.; Machado, M.; Silva, M.T.; Martins, J.M. On the Continuous Contact Force Models for Soft Materials in Multibody Dynamics. *Multibody Syst Dyn* **2011**, *25*, 357–375. [CrossRef]
19. Gonthier, Y.; McPhee, J.; Lange, C.; Piedbœuf, J.-C. A Regularized Contact Model with Asymmetric Damping and Dwell-Time Dependent Friction. *Multibody Syst. Dyn.* **2004**, *11*, 209–233. [CrossRef]
20. Gummer, A.; Sauer, B. Modeling Planar Slider-Crank Mechanisms with Clearance Joints in RecurDyn. *Multibody Syst. Dyn.* **2014**, *31*, 127–145. [CrossRef]
21. Shi, Y.; Rex, S.X.; Wang, X.; Hu, Z.; Newman, D.; Ding, W. Numerical Simulation and Field Tests of Minimum-Tillage Planter with Straw Smashing and Strip Laying Based on EDEM Software. *Comput. Electron. Agric.* **2019**, *166*, 105021. [CrossRef]
22. Ryan, R.R. ADAMS—Multibody System Analysis Software. In *Multibody Systems Handbook*; Schiehlen, W., Ed.; Springer: Berlin/Heidelberg, Germany, 1990; pp. 361–402. ISBN 978-3-642-50997-1.
23. Erez, T.; Tassa, Y.; Todorov, E. Simulation Tools for Model-Based Robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 4397–4404.
24. Kildashti, K.; Dong, K.; Yu, A. Contact Force Models for Non-Spherical Particles with Different Surface Properties: A Review. *Powder Technol.* **2023**, *418*, 118323. [CrossRef]
25. Choi, J.; Ryu, H.S.; Kim, C.W.; Choi, J.H. An Efficient and Robust Contact Algorithm for a Compliant Contact Force Model between Bodies of Complex Geometry. *Multibody Syst. Dyn.* **2010**, *23*, 99–120. [CrossRef]
26. Pan, J.; Chitta, S.; Manocha, D. FCL: A General Purpose Library for Collision and Proximity Queries. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, St Paul, MN, USA, 14–18 May 2012; pp. 3859–3866.
27. Zhou, Y.; Pan, Y.; Lei, T.; Chen, J. Robotic Simulator with High-Precision Perception of Contact Dynamics. In Proceedings of the 2023 IEEE International Conference on Development and Learning (ICDL), Macau, China, 9–11 November 2023; pp. 354–360.
28. Flores, P.; Lankarani, H.M. *Contact Force Models for Multibody Dynamics*; Solid Mechanics and Its Applications; Springer International Publishing: Cham, Switzerland, 2016; Volume 226, ISBN 978-3-319-30896-8.
29. Goldsmith, W. *Impact: The Theory and Physical Behaviour of Colliding Solids*; E. Arnold: London, UK, 1960.
30. Rodrigues Da Silva, M.; Marques, F.; Tavares Da Silva, M.; Flores, P. A Compendium of Contact Force Models Inspired by Hunt and Crossley's Cornerstone Work. *Mech. Mach. Theory* **2022**, *167*, 104501. [CrossRef]
31. Zhang, J.; Fang, M.; Zhao, L.; Zhao, Q.; Liang, X.; He, G. A Continuous Contact Force Model for the Impact Analysis of Hard and Soft Materials. *Mech. Mach. Theory* **2022**, *177*, 105065. [CrossRef]
32. Polyanin, A.D.; Zaitsev, V.F. *Handbook of Exact Solutions for Ordinary Differential Equations*, 2nd ed.; Chapman & Hall/CRC: Boca Raton, FL, USA, 2003; ISBN 978-1-58488-297-8.
33. Michel, L.; Ghanes, M.; Plestan, F.; Aoustin, Y.; Barbot, J.-P. Semi-Implicit Homogeneous Euler Differentiator for a Second-Order System: Validation on Real Data. In Proceedings of the 2021 60th IEEE Conference on Decision and Control (CDC), Austin, TX, USA, 13–15 December 2021; pp. 5911–5917.
34. Hongpan, N.; Yong, G.; Zhongming, H. Application Research of PhysX Engine in Virtual Environment. In Proceedings of the 2010 International Conference on Audio, Language and Image Processing, Shanghai, China, 23–25 November 2010; pp. 587–591.
35. Chang, H.-C.; Wang, J.-J. A Stochastic Grinding Force Model Considering Random Grit Distribution. *Int. J. Mach. Tools Manuf.* **2008**, *48*, 1335–1344. [CrossRef]
36. Open CASCADE Technology: Introduction. Available online: <https://dev.opencascade.org/doc/overview/html/index.html> (accessed on 28 January 2024).
37. Yang, P. Py0330/Arise. Available online: <https://github.com/py0330/aris> (accessed on 28 January 2024).

-
38. Tang, J.; Du, J.; Chen, Y. Modeling and Experimental Study of Grinding Forces in Surface Grinding. *J. Mater. Process. Technol.* **2009**, *209*, 2847–2854. [[CrossRef](#)]
 39. Zhao, H.; Wen, K.; Lei, T.; Xiao, Y.; Pan, Y. Automatic Aluminum Alloy Surface Grinding Trajectory Planning of Industrial Robot Based on Weld Seam Recognition and Positioning. *Actuators* **2023**, *12*, 170. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.