Interactive Simulation of Rigid Body Dynamics in Computer Graphics

Jan Bender[1], Kenny Erleben[2], Jeff Trinkle[3] and Erwin Coumans[4]

[1]Graduate School CE, TU Darmstadt, Germany
[2]Department of Computer Science, University of Copenhagen, Denmark
[3]Department of Computer Science, Rensselaer Polytechnic Institute, USA
[4]Advanced Micro Devices, Inc., USA

Abstract

Interactive rigid body simulation is an important part of many modern computer tools. No authoring tool nor a game engine can do without. The high performance computer tools open up new possibilities for changing how designers, engineers, modelers and animators work with their design problems.
This paper is a self contained state-of-the-art report on the physics, the models, the numerical methods and the algorithms used in interactive rigid body simulation all of which has evolved and matured over the past 20 years. The paper covers applications and the usage of interactive rigid body simulation.
Besides the mathematical and theoretical details that this paper communicates in a pedagogical manner the paper surveys common practice and reflects on applications of interactive rigid body simulation. The grand merger of interactive and off-line simulation methods is imminent, multi-core is everyman's property. These observations pose future challenges for research which we reflect on. In perspective several avenues for possible future work is touched upon such as more descriptive models and contact point generation problems. This paper is not only a stake in the sand on what has been done, it also seeks to give newcomers practical hands on advices and reflections that can give experienced researchers afterthought for the future.

Keywords: Rigid Body Dynamics, Contact Mechanics, Articulated Bodies, Jointed Mechanisms, Contact Point Generation, Iterative Methods.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.5]: Computational Geometry and Object Modeling—Physically-based modeling; Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism—Animation; Mathematics of Computing [G.1.6]: Numerical Analysis—Nonlinear programming

1. Motivation and Perspective on Interactive Rigid Body Simulation

Rigid body dynamics simulation is an integral and important part of many modern computer tools in a wide range of application areas like computer games, animation software for digital production including special effects in film and animation movies, robotics validation, virtual prototyping, and training simulators just to mention a few.

In this paper we focus on interactive rigid body dynamics simulation a subfield that has evolved rapidly over the past 10 years and moved the frontier of run-time simulation to applications in areas where off-line simulation only recently were possible. As a consequence this changes the computer tools humans use and has great social economical impact on society as a whole.

The term "interactive" implies a loop closed around a human and simulation tool. For applications like games where the feedback is simply animation on a screen, a reasonable goal is that the simulation deliver 60 frames per second (fps). For haptic rendering, the simulation would be part of a feedback loop running at 1000Hz, where this rate is needed to display realistic forces to the user.

In this state-of-the-art paper we will cover the important past 20 years of work on interactive rigid body simulation since the last state-of-the-art report [Bar93b] on the subject. Rigid body dynamics has a long history in computer graph-

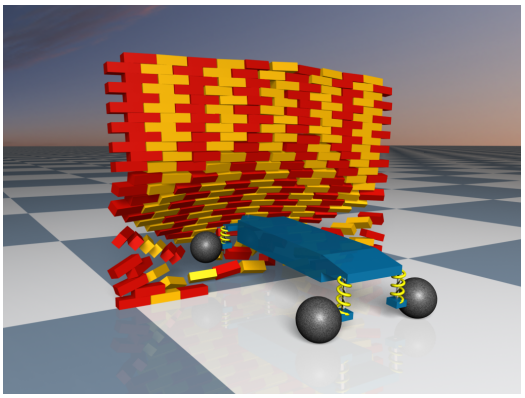


Figure 1: *Interactive rigid body simulations require the efficient simulation of joints, motors, collisions and contacts with friction.*

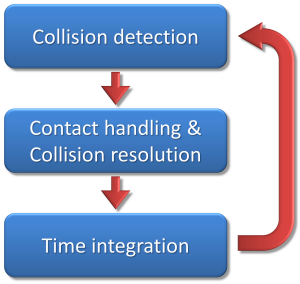


Figure 2: *The simulation loop provides a coarse description of data flow and processes in a rigid body simulator.*

ics for more than 30 years [AG85, MW88, Hah88, Bar89, BBZ91] and a wealth of work exists on the topic. As early as 1993 there were written state-of-the-art reports on the subject [Bar93b]. In his 93 STAR paper Baraff discussed penalty based methods and constraint based methods being an acceleration-level linear complementarity problem formulation. He did not cover many details on solving the linear complementarity problem. Not until 94 where Baraff published his version of a direct method based on pivoting was it feasible to compute solutions for Baraff's complementarity problem formulation. For years the 94 Baraff solution was the de-facto standard method of rigid body dynamics choice in both Maya and Open Dynamics Engine [Smi00]. However, the solution only remained interactive for small sized configurations (below 100 interacting objects or so). When the number of interacting objects increased the computational cost quickly made simulations last for hours and the acceleration-level formulation caused problems too with existence of solutions and uniqueness. Besides, solutions found by his algorithm did not always satisfy the static friction constraints. In the following years after Baraff's 1994 results, the impulse based paradigm was revisited by Mirtich in 96 [Mir96b] and become a strong competitor when concerning interactive simulation. Soon the interactive simulation community moved onto iterative methods and velocity level formulations, eventually evolving into the technology one finds today in engines such as Bullet [Cou05] and Open Dynamics Engine. As of this writing interactive simulation on single core CPUs with several 1000 and up to 10000 interacting objects are feasible. Multi-core and GPU works even go far beyond these limits. Even today much active cross-disciplinary work is ongoing on different contact formulations and iterative solvers taking in people not only from the field of computer graphics, but also from applied math, contact mechanics, robotics and more. Looking beyond contact problems, one also finds that simulation methods for articulated bodies have also undergone rapid development. In computer graphics, the reduced coordinate formulations have won much recognition as being superior for interactive rag-doll simulations.

1.1. The Anatomy and Physiology of a Rigid Body Simulator

A rigid body simulator is a complex and large piece of software. Traditionally it has been broken down into smaller well-defined pieces that each are responsible for solving a simpler single task. All pieces are tied together by a simulation loop shown in Figure 2. The loop begins with a collision detection query to find the contact points between the various bodies. These points are needed to write the physical laws governing the motions of the bodies, which are then solved to determine contact forces that provide proper contact friction effects and prevent bodies from interpenetrating. This phase is termed "contact handling." Newly formed contacts imply collisions, which are accompanied by impulsive forces (i.e., forces with infinite magnitudes over infinitesimal time periods). Impulsive forces cause instantaneous changes in the body velocities and so are often handled separately from pre-existing resting contacts. One refers to this as "collision resolving." After computing all the contact forces, the positions and velocities of the bodies are integrated forward in time before a new iteration of the simulation loop starts. Several iterations of the loop might be performed before a frame is rendered.

In order to derive the correct physical laws for the scene, all contacts between bodies must be found. If there are n bodies, then there are $O(n^2)$ pairs of bodies to test for collisions. To avoid collision detection becoming a computational bottleneck, it is broken into phases. In the first phase, called the "broad phase", bodies are approximated by simple geometric primitives for which distance computations are very fast (see Figure 3). For example, each body is replaced by the smallest sphere that completely contains it. If the spheres covering two bodies do not overlap, then neither do the actual bodies. The broad phase culling happens in global world coordinates. If the individual bodies are complex and

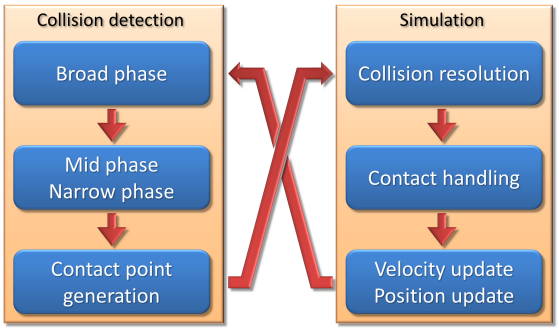


Figure 3: *A modular phase description of the sub tasks of a rigid body simulator helps decomposing a large complex system into simpler components.*

consist of many parts an additional stage called "mid phase" is used to cull parts in local body space. The culling is typically performed using bounding volume hierarchies. In the "narrow phase" the detailed geometries of bodies are used to find the precise body features in contact and the location of the contact points. However, this expensive operation must be done only for the pairs of bodies with overlapping approximations. The narrow phase is often mixed with the mid phase for performance reasons. Note that some narrow phase algorithms do not return all the required contact information, in which case a separate contact point generation algorithm can be applied (see section 6.3)

1.2. The Quest for Robustness, Accuracy, and Performance

The recent trend in interactive rigid body simulation has focused on delivering larger and larger simulations of rigid bodies or creating simulation methods that can deliver results faster. Thus, the old saying *bigger and faster is better* is very descriptive for many past works in the field of interactive rigid body simulation. The need for bigger and faster is motivated by rigid body simulators being used in for instance digital production. It looks more interesting to have a pile of skeleton skulls in a movie than having a hand full of cubes and spheres. Thus, the need in production for creating interesting motion requires more complex simulation scenarios.

The well known tradeoff between accuracy and performance is an inherent property of interactive rigid body simulation. Many applications enforce a performance constraint which leaves too little time for computing accurate solutions. Thus, one must often balance accuracy and stability properties to meet the performance constraint.

Robustness is another desirable numerical trait of a simulator. The motivation for this is often caused by having a human being (or the real world in case of robotics) interacting with a simulator. This may be the cause for much pain and frustrations as humans have a tendency to be unpredictable.

In summary, the holy grail of interactive rigid body simulation is extremely fast and robust simulation methods that can deal gracefully with large scale complex simulation scenarios under hard performance constraints.

1.3. Application Areas of Interactive Rigid Body Simulation

The maturing technology makes it possible to use rigid body simulators as sub-parts in larger systems. For instance in time critical scenarios like tracking humans or maneuvering a robot, a simulator can be used as a prediction tool.

From a digital design viewpoint, one may define a spectrum of technology. At one end of the spectrum one finds off-line simulators that may take hours or days to compute results, but on the other hand they deliver high quality results. For movie production several such computer graphics simulation methods have been presented [Bar94, GBF03, KSJP08]. At the other end of the spectrum one finds the fast run-time simulators capable of delivering plausible results very fast. This kind of simulator often originates from game physics. One example is Bullet. At the middle of the spectrum one finds moderately fast simulators that may deliver high fidelity results. These are very suitable for testing design ideas or training.

In general different application areas have different needs in regards to performance/quality trade-offs and accuracy. With this in mind we will discuss a few application areas in the next four subsections.

1.3.1. Entertainment for Games and Movies

For games and movies rigid body simulation has to be plausible rather than physically realistic. For games, the simulation needs to be real-time. Simulations for movies do not have the real-time constraint, but fast simulation methods are also preferred, since very complex scenarios are simulated for special effects and simulation time costs money. Therefore, the development in the two areas go in the same direction. Iterative constraint solving methods are popular in both areas.

Many games using 2D and 3D graphics rely on a rigid body dynamics engine to deal with collision detection and collision response. In some cases the motion of the objects is fully driven by rigid body dynamics, for example the game Angry Birds,using the Box2D physics engine or a 3D Jenga game. More commonly, object motion is customized in a non-physical way to a certain degree, to favor a satisfying game playing experience over physical realism. This introduces the challenge of interaction between rigid bodies and kinematically animated objects. Kinematically animated objects can be represented as rigid bodies with infinite mass,

so that the interaction is one way. The influence from rigid body to kinematically animated objects is often scripted in a non-physical way.

With increasing CPU budgets, there is growing interest in using more realistic, higher quality simulation. In particular the combination of rag-doll simulation, animation, inverse kinematics and control requires better methods. It requires constraint solvers that can deal with very stiff systems and strong motors that can deal with the large change in velocity. Several game and movie studios are using Featherstone's articulated body method to simulate rag-dolls.

Destruction and fracture of objects can generate a lot of dynamic rigid bodies, and to handle them, games use multicore CPUs or offload the rigid body simulation onto GPUs.

1.3.2. Interactive Digital Prototyping

Interactive virtual prototyping can be an important computer tool for verifying a design idea or as a pre-processing tool to tune parameters for more computational expensive simulation tools. CEA LIST [CEA11], CMLabs, robot simulators Webots, Gazebo, or Microsoft robotics developer studio [KP09, Cyb09, Mic09] are a few examples of many such tools. The main goal is to reduce the time to market and thereby lower overall production costs. A secondary goal is development of better products of higher quality.

Interactive prototyping has been motivated by the computational fast technology that has evolved in the gaming and movie industries. The instant feedback that can be obtained from such simulations is attractive for rapid iterative prototyping. However, although interactivity is attractive, one can not compromise the physical correctness too much. Thus, plausible simulation [BHW96] may not be good enough for trusting a virtual design. A current trend is seen where interactive simulation tools are improved for accuracy and moved into engineering tools [Stu08, TNA08, TNA*10, CA09].

Even the European space agency (ESA) is using PhysX for verification of the Mars sample rover for the ExoMars Programme to investigate the Martian environment [KK11].

1.3.3. Robotics

The main goal of the field of robotics is the development of intelligent man-made physical systems that can safely and efficiently accomplish a wide range of tasks that aid the achievement of human societal goals. Tasks that are particularly difficult, dangerous or boring are good candidates for robotic methods, e.g., assembly in clean-room environments, extra-terrestrial exploration, radioactive materials handling, and laparoscopic surgery. In addition, methods for robotics are increasingly being applied in the development of new generations of active prosthetic devices, including hands, arms and legs. The most challenging of these tasks most directly related to this paper are those that cannot be accomplished without (possibly) intermittent contact, such as walking, grasping and assembly.

Until now, robot manipulation tasks involving contact have been limited to those which could be accomplished by costly design of a workcell or could be conducted via teleoperation. The ultimate goal, however, is to endow robots with an understanding of contact mechanics and task dynamics, so they can reason about contact tasks, automatically plan and execute them, and enhance their manipulation skills through experience. The fundamental missing component has been fast, physical simulation tools that accurately model effects such as stick-slip friction, flexibility, and dynamics. Dynamics is important for robots to perform manipulation tasks quickly or to allow it to run over uneven terrain. Physical simulation today has matured to the point where it can be integrated into algorithms for robot design, task planning, state-estimation and control. As a result, the number of robotic solutions to problems involving contact is poised to experience a major acceleration.

1.3.4. Industrial and Training Simulators

Computer simulators are cheap and risk free ways to train people to handle heavy equipment in critical situations under large stress. Examples include large forest machines as well as bulldozers to cable simulations in tug boats and maneuver belt vehicles [SL08]. Driving simulators are another good example [Uni11, INR11]. The simulators in this field need to be responsive as well as accurate enough to give proper predictions of the virtual equipment being handled. This is similar to interactive virtual prototyping. In fact in our view it is mostly the purpose that distinguishes the two, one is design the other is training.

The driving simulators are very specialized and include many aspects of virtual reality. The largest and most complex simulator, National Advanced Driving Simulator (NADS), costs in the order of $54 million. In contrast, Gazebo is free software and commercial software such as Vortex [CM 11] or Algoryx [Alg11]) are cheap in comparison with NADS.

2. A Quick Primer

Rigid body simulation is analogous to the numerical solution of nonlinear ordinary differential equations for which closed-form solutions do not exist. Assume time t is the independent variable. Given a time period of interest $[t_0, t_N]$, driving inputs, and the initial state of the system, the differential equations (the instantaneous-time model) are discretized in time to yield an approximate discrete-time model, typically in the form of a system of (state-dependent) algebraic equations and inequalities. The discrete-time model is formulated and solved at each time of interest, $(t_0, .., t_N)$. In rigid body simulation, one begins with the Newton-Euler (differential) equations, which describes the dynamic motion

of the bodies without contact. These differential equations are then augmented with three types of conditions: nonpenetration constraints that prevent the bodies from overlapping, a friction model that requires contact forces to remain within their friction cones, and complementarity (or variational inequality) constraints that enforce certain disjunctive relationships among the variables. These relationships enforce critically important physical effects; for example, a contact force must become zero if two bodies separate and if bodies are sliding on one another, the friction force acts in the direction that will most quickly halt the sliding. Putting all these components together yields the instantaneous-time model, as a system of differential algebraic equations and inequalities that can be reformulated as a differential nonlinear complementarity problem (dNCP). The dNCP cannot be solved in closed form or directly, so instead, one discretizes it in time, thereby producing a sequence of NCPs whose solutions approximate the state and contact force trajectories of the system. In the ideal case, the discrete trajectories produced in this process will converge trajectories of the original instantaneous-time model. Computing a discrete-time solution requires one to consider possible reformulations of the NCPs and a choice of solution method. There are many options for instance reformulation as nonsmooth equation using Fischer-Burmeister function or proximal point mappings etc.

2.1. Classical Mechanics

Simulation of the motion of a system of rigid bodies is based on a famous system of differential equations, the *Newton-Euler equations*, which can be derived from Newton's laws and other basic concepts from classical mechanics:

- Newton's 1st law: The velocity of a body remains unchanged unless acted upon by a force.
- Newton's 2nd law: The time rate of change of momentum of a body is equal to the applied force.
- Newton's 3rd law: For every force there is an equal and opposite force.

Two important implications of Newton's laws when applied to rigid body dynamics are: (from the first law) the equations apply only when the bodies are observed from an inertial (non-accelerating) coordinate frame and (from the third law) at a contact point between two touching bodies, the force applied from one body onto the second is equal in magnitude, opposite in direction, and collinear with the force applied by the second onto the first. Applying these two implications to Newton's second law gives rise to differential equations of motion. While the second law actually applies only to particles, Euler was kind enough to extend it to the case of rigid bodies by viewing them as collections of infinite numbers of particles and applying a bit of calculus [GPS02, ESHD05]. This is why the equations of motion are known as the Newton-Euler equations.

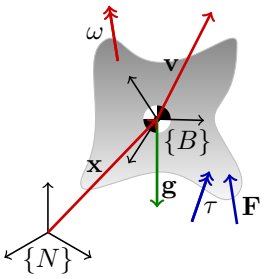


Figure 4: *Illustration of a spatial rigid body showing the body frame $\{B\}$ and inertial frame $\{N\}$ as well as notation for positions, velocities and forces.*

Before presenting the Newton-Euler equations, we need to introduce a number of concepts from classical mechanics. Figure 4 shows a rigid body in space, moving with translational velocity $\mathbf{v}$ and rotational velocity ω, while being acted upon by an applied force $\mathbf{F}$ and moment τ (also known as a torque).

2.1.1. Rigid Bodies

A *rigid body* is an idealized solid object for which the distance between every pair of points on the object will never change, even if huge forces are applied. A rigid body has mass m, which is distributed over its volume. The centroid of this distribution (marked by the circle with two blackened quarters) is called the center of mass. To compute rotational motions, the mass distribution is key. This is captured in a 3×3 matrix known as the mass (or inertia) matrix $\mathbf{I} \in \mathbb{R}^{(3\times3)}$. It is symmetric and positive definite matrix with elements known as moments of inertia and products of inertia, which are integrals of certain functions over the volume of the body [Mei70]. When the integrals are computed in a body-fixed frame, the mass matrix is constant and will be denoted by $\mathbf{I}_{\text{body}}$. The most convenient body-fixed frame for simulation is one with its origin at the center of mass and axes oriented such that $\mathbf{I}_{\text{body}}$ is diagonal. When computed in the inertial frame, the mass matrix is time varying and will be denoted by $\mathbf{I}$.

2.1.2. Rigid Body Kinematics

The body's position in the inertial (or world) frame is given by the vector $\mathbf{x} \in \mathbb{R}^3$, from the origin of the inertial frame $\{N\}$ fixed in the world to the origin of the frame $\{B\}$ fixed in the body. Note that since three independent numbers are needed to specify the location of the center of mass, a rigid body has three translational degrees of freedom.

The orientation of a rigid body is defined as the orientation of the body-fixed frame with respect to the inertial frame. While many representations of orientation exist, here we use rotation matrices $\mathbf{R} \in \mathbb{R}^{3\times3}$ and unit quaternions

$Q \in \mathbb{H}$. Rotation matrices are members of the class of *orthogonal matrices*. Denoting the columns by $\mathbf{R}_1$, $\mathbf{R}_2$, and $\mathbf{R}_3$, orthogonal matrices must satisfy: $\| \mathbf{R}_i \| = 1$; $i = 1,2,3$ and $\mathbf{R}_i^T \mathbf{R}_j = 0$; $\forall i \neq j$; $i = 1,2,3$; $j = 1,2,3$. Since the nine numbers in $\mathbf{R}$ must satisfy these six equations, only three numbers can be freely chosen. In other words, a rigid body has three rotational degrees of freedom. A unit quaternion is four numbers $[Q_s,\ Q_x,\ Q_y,\ Q_z]$, constrained so that the sum of their squares is one. The fourth element can be computed in terms of the other three, and this redundancy serves as additional confirmation that orientation has three degrees of freedom. Considering translation and rotation together, a rigid body has six degrees of freedom.

The *rotational velocity* $\omega \in \mathbb{R}^3$ (also known as, *angular velocity*) of a body can be thought of as vector whose direction identifies a line about which all points on the body instantaneously rotate (shown as a red vector with a double arrowhead in Figure 4). The magnitude determines the rate of rotation. While the rate of rotation may be changing over time, at each instant, every point on a rigid body has exactly the same rotational velocity. The three elements of ω correspond to the three rotational degrees of freedom.

Translational velocity $\mathbf{v} \in \mathbb{R}^3$ (also inaccurately referred to as *linear velocity*) is an attribute of a point, not a body, because when a body rotates, not all points have the same velocity (see the red vector with a single arrowhead in Figure 4). However, the velocity of every point can be determined from the velocity of one reference point and the angular velocity of the body. In rigid body dynamics, the center of mass is typically chosen as the reference point.

Next we need velocity *kinematic* relationships. Kinematics is the study of motion without concern for forces, moments, or body masses. By contrast, *dynamics* is the study of how forces produce motions. Since dynamic motions must also be kinematically feasible, kinematics is an essential building block of dynamics. The particular kinematic relationships needed here relate the time derivatives of position and orientation variables to the translational and rotational velocities.

Let us define $\mathbf{q} = (\mathbf{x},\ Q)$ as the tuple containing the position of the center of mass and the orientation parameters. Note that the length of $\mathbf{q}$ is seven if Q is a quaternion (which is the most common choice). The generalized velocity of the body is defined as: $\mathbf{u} = [\mathbf{v}^T\ \omega^T]^T \in \mathbb{R}^6$. The velocity kinematic equations for a rigid body relate $\dot{\mathbf{q}}$ to $\mathbf{u}$, which may have different numbers of elements. The relationship between the translational quantities is simple: $\dot{\mathbf{x}} = \mathbf{v}$. The time rate of change of the rotational parameters Q is a bit more complicated; it is the product of a Jacobian matrix and the rotational velocity of the body: $\dot{Q} = \mathbf{G}(Q)\omega$, where the details of $\mathbf{G}(Q)$ are determined by the orientation representation. In the specific case when Q is a unit quaternion, $\mathbf{G}(Q)$ is defined as follows:

$$\mathbf{G} = \frac{1}{2}\begin{bmatrix} -Q_x & -Q_y & -Q_z \\ Q_s & Q_z & -Q_y \\ -Q_z & Q_s & Q_x \\ Q_y & -Q_x & Q_s \end{bmatrix}.$$

Putting the two velocity kinematic relationships together yields:

$$\dot{\mathbf{q}} = \mathbf{H}\mathbf{u} \tag{1}$$

where $\mathbf{H} = \begin{bmatrix} \mathbf{1}_{3\times 3} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} \end{bmatrix}$, where $\mathbf{1}_{3\times 3}$ is the 3-by-3 identity matrix. Note that when the orientation representation uses more than three parameters, $\mathbf{G}$ is not square, although it has the property that $\mathbf{G}^T\mathbf{G} = \mathbf{1}$, where $\mathbf{1}$ is the identity matrix of size 3.

2.1.3. Constraints

Constraints are equations and inequalities that change the way pairs of bodies are allowed to move relative to one another. Since they are kinematic restrictions, they also affect the dynamics. Constraints do *not* provide a direct means to compute the forces that must exist to enforce them. Generally, constraints are functions of generalized position variables, generalized velocities, and their derivatives to any order:

$$C(\mathbf{q}_1, \mathbf{q}_2, \mathbf{u}_1, \mathbf{u}_2, \dot{\mathbf{u}}_1, \dot{\mathbf{u}}_2, ..., t) = 0 \tag{2}$$

or

$$C(\mathbf{q}_1, \mathbf{q}_2, \mathbf{u}_1, \mathbf{u}_2, \dot{\mathbf{u}}_1, \dot{\mathbf{u}}_2, ..., t) \geq 0 \tag{3}$$

where the subscripts indicate the body. Equality and inequality constraints are referred to as *bilateral* and *unilateral* constraints, respectively.

As an example, consider two rigid spheres of radii r_1 and r_2 and with centers located at $\mathbf{x}_1$ and $\mathbf{x}_2$. Consider the constraint function:

$$C(\mathbf{x}_1, \mathbf{x}_2) = ||\mathbf{x}_1 - \mathbf{x}_2|| - (r_1 + r_2),$$

where $||\cdot||$ is the Euclidean two-norm. If $C = 0$, then the surfaces of the spheres touch at a single point. If this bilateral constraint is imposed on the Newton-Euler equations, then regardless of the speeds of the spheres and the sizes of the forces, the surfaces will always remain in single-point contact. Intuitively, for this to happen the constraint force normal to the sphere surfaces can be compressive (the spheres push on each other) or tensile (the spheres pull). By contrast, $C \geq 0$, then the two spheres may move away from each other but never overlap. Correspondingly, the constraint force can only be compressive.

The form of a constraint (see Figure 5) impacts the way in which the Newton-Euler equations should be solved. *Holonomic* constraints are those which can be expressed as an equality in terms of only generalized position variables and time. These are further subdivided into those independent of

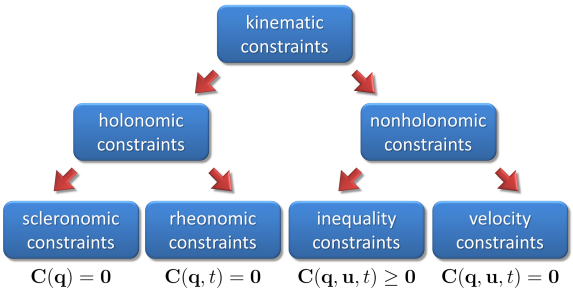


Figure 5: *Constraint classification*

time, known as *scleronomic*, and those dependent on time, *rheonomic*. An example of a scleronomic constraint is the equality constraint of the spheres discussed above). Rheonomic constraints typically arise when one body is kinematically controlled (*i.e.*, it is required to follow a known trajectory regardless of the forces that might be required to make that happen).

Any constraint that is not holonomic is said to be *nonholonomic*. This class includes all unilateral constraints and equality constraints which are not integrable in the sense that generalized velocity variables and derivatives of the generalized position variables (and higher derivatives, if present) cannot be eliminated. The steering constraint for a car on a flat surface whose wheels are not allowed to skid is a nonholonomic equality constraint. If the car is driving along, then its rotational velocity is directly proportional the car's forward speed and the angle of the front wheels. This means the fundamental constraint between two velocities cannot be integrated to yield an equivalent constraint written solely in terms of position variables, hence the constraint is nonholonomic.

Holonomic constraints remove degrees of freedom from the system, *i.e.,* the dimension of the space of possible generalized positions is reduced. For instance two free rigid bodies have a total of 12 degrees of freedom, but as in the previous case of the touching spheres, one degree of freedom is lost. Assume that one sphere can be moved at will through space using all six degrees of freedom. Now view the second sphere from a frame of reference fixed in the first. From this perspective, the second sphere can rotate with all three degrees of freedom while maintaining contact and also translate with the contact point moving across the surface of the first sphere. Since this surface is two-dimensional, the second sphere has only two translational degrees of freedom. Thus a system of two spheres with one contact constraint has 11 degrees of freedom. If instead, two bodies were connected by a hinge joint, the system would have seven degrees of freedom. That is, if you allow one body to move with six degrees of freedom, then the other can only rotate about the hinge joint with respect to the first body. This also implies that a hinge constraint cannot be represented with fewer than five holonomic constraints.

One should note that non-holonomic equality constraints remove only instantaneous, or local, degrees of freedom from the system. In the car example, the car cannot translate instantaneously directly left or right. However, every competent driver can accomplish a lateral move of his car by executing the kind of maneuver used to parallel park in a small space.

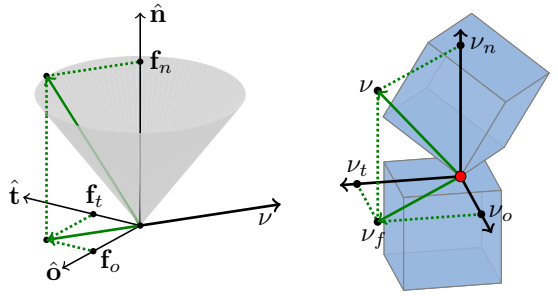
2.1.4. Forces and Moments and Relative Velocity

A *force* $\mathbf{f}$ is a vector with a line of action. A force produces a *moment* τ or *torque* about any point not on the line of action of the force. Let $\mathbf{r}$ and ρ be two distinct points such that $\mathbf{r}$ is on the line of action and ρ is not. Then the moment of $\mathbf{f}$ with respect to $\mathbf{r}$ is defined as $\tau = (\mathbf{r} - \rho) \times \mathbf{f}$. Moments need not be byproducts of forces; they exist in their own right, which is why one is shown applied to the body in Figure 4.

Many sources of forces exist in rigid body dynamics, for example, forces from wind, gravity, and electro-magnetics. However, the forces that are most difficult to deal with, but also critically important in interactive simulation are constraint and friction forces.

Gravity, as we experience it on Earth, acts equally on every particle of mass in a rigid body. Nonetheless, the gravity force is shown in Figure 4 as a single force of magnitude mg with line of action through the center of mass of the body. This is because the affect of gravity acting on an entire body is equivalent to a single force of magnitude mg acting through its center of mass. Friction forces are dissipative. They act in contact interfaces to halt sliding at sliding contacts and to prevent sliding at sticking and rolling contacts. The type of friction force focused on here is *dry friction*, which is assumed to act at contacts between body surfaces, including the inner surfaces of joints. Dry friction, as opposed to viscous friction, allows bodies to stick together and requires a non-zero tangential force to initiate sliding.

For point contacts between body surfaces, we consider the standard isotropic Coulomb friction model. Assume that contact occurs at a single point with a uniquely defined tangent plane. Then place the origin of the contact coordinate frame at the contact point and let the t- and o-axes lie in the tangent plane (see Figure 6(a)). The n-axis is orthogonal to the t- and o-axes and is referred to as the contact normal. A contact force $\mathbf{f}$ is decomposed into a normal componen $\mathbf{f}_n$ and tangential components, $\mathbf{f}_t$ and $\mathbf{f}_o$. Because bodies are able to push against each other, but not pull, the normal force is unilateral, *i.e.,* $\mathbf{f}_n \geq 0$. Similarly, the relative velocity between the touching points on the bodies ν is decomposed into components, ν_n, ν_t, and ν_o (see Figure 6(b)). The contact is sliding if $\nu_n = 0$ and ν_t or ν_o is nonzero, and separating if ν_n is greater than zero. Negative ν_n is not allowed, as it corresponds to interpenetration of the bodies.



(a) A Friction Cone (b) Contact velocities

Figure 6: *The friction cone of a contact and the decomposition of the relative contact velocity.*

The Coulomb model has two conditions: first, the net contact force must lie in a quadratic friction cone (see the gray cone in Figure 6(a)) and second, when the bodies are slipping, the friction force must be the one that directly opposes sliding. The cone is defined as follows:

$$\mathcal{F}(\mathbf{f}_n,\mu) = \{\mu^2\mathbf{f}_n^2 - \mathbf{f}_t^2 - \mathbf{f}_o^2 \geq 0, \mathbf{f}_n \geq 0\} \tag{4}$$

where $\mu \geq 0$ is the friction coefficient. The friction force that maximizes friction dissipation is:

$$\mathbf{f}_t = -\mu\mathbf{f}_n\frac{\nu_t}{\beta} \tag{5}$$

$$\mathbf{f}_o = -\mu\mathbf{f}_n\frac{\nu_o}{\beta} \tag{6}$$

where $\beta = \sqrt{\nu_t^2+\nu_o^2}$ is the sliding speed at the contact (see Figure 6(b)).

Common variations on this model include using two different friction coefficient; one for sticking contact and a lower one for sliding. When friction forces are higher in one direction than another, one can replace the circular cone with an elliptical cone. In some simulation schemes the nonlinearity of the friction cone causes problems, and so it is eliminated by approximating the cone as a symmetric polyhedral cone. Finally, to model the fact that contacts between real bodies are actually small patches, the friction cone can be extended, as done by Contensou, to allow for a friction moment that resists rotation about the contact normal [Con93, TTP01].

A similar model for dry friction acting to resist joint motion will be discussed in section 3.

2.1.5. The Newton-Euler Equations

The Newton-Euler equations are obtained by applying Newton's second law twice; once for translational motion and again for rotational motion. Specifically, the net force $\mathbf{F}$ applied to the body is equal to the time rate of change of translational momentum $m\mathbf{v}$ (*i.e.*, $\frac{d}{dt}(m\mathbf{v}) = \mathbf{F}$) and the net moment τ is equal to the time rate of change of rotational momentum $\mathbf{I}\omega$ (*i.e.*, $\frac{d}{dt}(\mathbf{I}\omega) = \tau$). Specializing these equations to the case of a rigid body (which, by definition, has constant mass) yields:

$$m\dot{\mathbf{v}} = \mathbf{F} \tag{7}$$

$$\mathbf{I}\dot{\omega} + \omega \times \mathbf{I}\omega = \tau. \tag{8}$$

where recall that $\mathbf{I}$ is the 3-by-3 inertia matrix and $\times$ represents the vector cross product.

The second term on the left side of the rotational equation is called the "gyroscopic force" which arises from the proper differentiation of the rotational momentum. The rotational velocity and mass matrix must both be expressed in the same frame, which is usually taken as a body-fixed frame (which is rotating with the body in the inertial frame) or the inertial frame. In a body-fixed frame, $\mathbf{I}_{body}$ is constant, but ω is a vector expressed in a rotating frame, which means that $\mathbf{I}\omega$ is also a vector expressed in a rotating frame. The first term represents the rate of increase of angular velocity along the vector ω.

One might be tempted to try to eliminate the second term by expressing the rotational quantities in the inertial frame and differentiating them there. However this does not work, because the inertia matrix expressed in the inertial frame $\mathbf{I}$ is time-varying, as seen by the following identity $\mathbf{I} = \mathbf{R}\mathbf{I}_{\text{body}}\mathbf{R}^T$. Differentiating inertial frame quantities yields an equivalent expression with equivalent complexity.

The Newton-Euler equations contain the net force $\mathbf{F}$ and moment τ. $\mathbf{F}$ is simply the vector sum of all forces acting on the body. τ is the vector sum of the moments of all the forces and pure moments. One can see from equation (7), that the net force causes the center of gravity to accelerate in the direction of the net force proportional to its magnitude. This is true independent of the location of the line of action in space. Equation (8) implies that the net moment directly affects the rotational velocity of the body, but in a more complicated way. The gyroscopic moments tend to cause the axis of rotation of a rotating rigid body to "precess" about a circular cone.

Simulation of free body motion is done by integrating the Newton-Euler equations (7,8) and the velocity kinematic equation (1) simultaneously. If there are contacts and joints, then these equations must be augmented with the constraint equations (2,3). If in addition, dry friction exists in contacts, then equations (4,5,6) must be included. The complete system of differential and algebraic equations and inequalities is challenging to integrate, but methods to do this robustly have been developed over the past 20 years. To push the boundaries of interactive rigid body dynamics, one must maintain the current level of solution robustness and greatly increase the solution speed.

2.1.6. Impulse

When a pair of bodies collides, those bodies, and any other bodies they are touching, experience very high forces of very

short duration. In the case of ideal rigid bodies, the force magnitudes become infinite and the duration becomes infinitesimal. These forces are referred to as *impulsive forces* or *shocks*. One can see from equation (7), that shocks cause infinite accelerations, which makes direct numerical integration of the Newton-Euler equations impossible. One way to deal with this problem during simulation is to use a standard integration method up to the time of impact, then use an impulse-momentum law to determine the jump discontinuities in the velocities, and finally restart the integrator.

Let $[t, t+\Delta t]$ be a time step during which a collision occurs. Further, define $\mathbf{p} = \int_t^{t+\Delta t} \mathbf{F} dt$ as the impulse of the net force and $m\mathbf{v}$ as translational momentum. Integrating equation (7) from t to $t+\Delta t$ yields $m(\mathbf{v}(t+\Delta t) - \mathbf{v}(t)) = \int_t^{t+\Delta t} \mathbf{F} dt$, which states that impulse of the net applied force equals the change of translational momentum of the body. In rigid body collisions, Δt approaches zero. Taking the limit as Δt goes to zero, one obtains an impulse momentum law that is applied at the instant of impact to compute post collision velocities. Since Δt goes to zero and the velocities remain finite, the generalized position of the bodies are fixed during the impact. After processing the collision, one has the values of the generalized positions and velocities, which are the needed initial conditions to restart the integrator. Note that integration of the rotational equation (8) yields an impulse-momentum law for determining jump discontinuities in the rotational velocities.

Based on impulse-momentum laws, several algebraic collision rules have been proposed. Newton's Hypothesis is stated in terms of the normal component of the relative velocity of the colliding points just before and just after collision: $\mathbf{v}_n^+ = -\varepsilon \mathbf{v}_n^-$, where $\mathbf{v}_n^-$ is relative normal velocity just before impact, $\mathbf{v}_n^+$ is the relative normal velocity just after impact, and $\varepsilon \in [0,1]$ is known as the coefficient of restitution. Setting ε to zero yields a perfectly plastic impact (*i.e.*, an impact with no bounce). Setting this value to 1 yields perfectly elastic impacts (*i.e.*, no energy is lost).

Poisson's Hypothesis is similar, but is a function of collision impulse rather than the rate of approach. The normal impulse is divided into two parts, $\mathbf{p}_n^c$ and $\mathbf{p}_n^r$, which are related as follows $\mathbf{p}_n^r = \varepsilon \mathbf{p}_n^c$, where again $\varepsilon \in [0,1]$. Immediately prior to the collision, v_n^- of the impact points is negative. The compression impulse $\mathbf{p}_n^c$ is defined as the amount of impulse required to cause the relative normal velocity to become zero - just enough to prevent body interpenetration with no bounce. The restitution impulse is applied after the compression impulse to generate bounce (*i.e.*, $\mathrm{v}_n^- > 0$).

The same idea can be applied to frictional collision impulses by replacing the normal components of the impulses and velocities with the tangential components (see for example [Bra91]). The normal and tangential impact hypotheses can be used together to determine the velocity jumps caused by impacts. While simple and intuitive, this approach can unfortunately generate energy during oblique collisions. To prevent such unrealistic outcomes, Stronge developed an energy-based collision law that imposes a condition that prevents energy generation. Chatterjee and Ruina incorporated Stronge's energy constraint and recast the collision law in terms of two parameters that are physically meaningful [CR98].

3. Models for Interactive Simulation

The laws of physics must be combined into what we term an instantaneous-time model, which describes the continuous motions of the rigid bodies. Following this, we discretize this model over the time domain to obtain a discrete-time model, which is a sequence of so-called time-stepping subproblems. The subproblems are formulated and numerically solved at every time step to simulate the system.

In this section, we present generic models for systems with multiple simultaneous frictional contacts in Section 3.1. The particulars of models for dealing with aspects of reduced coordinate formulations are covered in Section 3.2.

3.1. Modeling of Simultaneous Frictional Contacts

Here we take a strict approach trying to keep the physics as correct as possible by only introducing errors of linearization and discretization. The model consists of five parts: the Newton-Euler equation [Lan86], a kinematic map (to relate time derivatives of configuration parameters to translational and angular velocity variables), equality constraints (to model permanent joint connections), normal contact conditions (to model intermittent contact behavior), and a dry friction law satisfying the principle of maximum power dissipation, also known as the principle of maximum work [Goy89]. These five parts will be explained in detail below.

Two types of constraints exist: permanent mechanical joints, each represented by a system of equations (five scalar equations in the case of a one-degree-of-freedom joint), and isolated point contacts with well-defined contact normals, each represented by one scalar inequality constraint. Let $\mathcal{B}$ and $\mathcal{U}$ denote the mutually exclusive sets of bilateral (equality) and unilateral (inequality) contacts:

$$\mathcal{B} = \{i : \text{contact } i \text{ is a joint}\} \tag{9}$$

$$\mathcal{U} = \{i : \text{contact } i \text{ is a point contact}\} \tag{10}$$

where $\mathcal{B} \cup \mathcal{U} = \{1, ..., n_c\}$ and n_c is the number of contacts. Note that distributed contacts can be approximated arbitrarily well by a number of isolated point contacts.

To formulate the equations of motion properly, one needs precise definitions of contact maintenance, sliding, and rolling. It is convenient to partition possible relative motions at each contact into *normal* and *frictional* subspaces. Let ${}^{\kappa}\mathbf{C}_{in}$ and ${}^{\kappa}\mathbf{C}_{if}$, where $\kappa \in \{b, u\}$, denote signed distance

functions (or gap functions) in the normal and friction subspace directions at contact i. If two bodies touch at contact i, then ${}^{\kappa}\mathbf{C}_{in} = 0$. This is always enforced for joints (${}^{b}\mathbf{C}_{in} = 0$), which are permanent contacts, but not for unilateral contacts, which can be broken as bodies separate (${}^{u}\mathbf{C}_{in} > 0$).

The first time derivatives of the distance functions are the relative contact velocities, ${}^{\kappa}\mathrm{v}_{i\sigma} = \frac{d}{dt}\left({}^{\kappa}\mathbf{C}_{i\sigma}\right)$; $\kappa \in \{b,\, u\}, \sigma \in \{n,\, f\}$. Note that ${}^{\kappa}\mathrm{v}_{in}$ and ${}^{\kappa}\mathrm{v}_{if}$ are orthogonal subspaces, where unallowed motions are prevented by body structures and sliding motions are resisted by friction forces, respectively. If a pair of contact points (one on each body at the point of touching) are in rolling contact, instantaneously, the distance between those bodies in the direction of possible sliding is zero (${}^{\kappa}\mathrm{v}_{if} = 0$). If they slip, at least one friction direction displacement will become nonzero. For example, the friction direction of a one-degree-of-freedom joint is in the direction of motion of the joint. For a unilateral contact with isotropic Coulomb friction, the friction subspace will consist of relative translation in the t- and o-directions. The corresponding displacement functions will be denoted by ${}^{u}\mathbf{C}_{it}$ and ${}^{u}\mathbf{C}_{io}$. Relative rotations are not resisted by body structure or friction, so they are not included in either subspace.

We now partition all contacts into sliding and rolling subsets. At the position level, contact i is sustained if the distance function ${}^{\kappa}C_{in}(\mathbf{q},t)$; $\kappa \in \{b,u\}$ is equal to zero for a finite period of time. However, one cannot distinguish sliding from rolling with this position-level condition; one needs time derivatives. The velocity-level set definitions are:

$$\mathcal{S} \;=\; \{i : {}^{\kappa}\mathbf{C}_{in} = 0,\ {}^{\kappa}\mathrm{v}_{in} = 0,\ {}^{\kappa}\mathrm{v}_{if} \neq 0\} \tag{11}$$

$$\mathcal{R} \;=\; \{i : {}^{\kappa}\mathbf{C}_{in} = 0,\ {}^{\kappa}\mathrm{v}_{in} = 0,\ {}^{\kappa}\mathrm{v}_{if} = 0\}, \tag{12}$$

where the sets $\mathcal{S}$ and $\mathcal{R}$ are mutually exclusive.

We are now in a position to develop the system of equations and inequalities defining the instantaneous-time dynamic model of a multi-rigid-body system with bilateral and unilateral contacts. Recall the five parts mentioned above.

Newton-Euler Equations: The Newton-Euler equation can be written as follows:

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{u}} = \mathbf{g}(\mathbf{q},\mathbf{u},t), \tag{13}$$

where $\mathbf{M}(\mathbf{q})$ is the generalized mass matrix containing the body mass properties and $\mathbf{g}(\mathbf{q},\mathbf{u},t)$ is the vector of loads, including the gyroscopic moment (the cross-product term in equation (15)). Specifically for the j^{th} rigid body we have:

$$\mathbf{M}_j = \begin{bmatrix} m_j\mathbf{1}_{3\times 3} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_j(Q_j) \end{bmatrix}, \tag{14}$$

$$\mathbf{g}_j = \begin{bmatrix} \mathbf{F}_j \\ \tau_j - \omega_j \times \mathbf{I}_j(Q_j)\omega_j \end{bmatrix}, \tag{15}$$

where $\mathbf{1}_{3\times 3}$ is the 3-by-3 identity matrix. Recall that $\mathbf{I}$ is the 3-by-3 inertia matrix, and $\mathbf{F}_j$ and τ_j are the externally applied force and moment. Also note that $\mathbf{M}$ is positive definite and symmetric.

Kinematic Map: The time rate of change of the generalized coordinates of the bodies $\dot{\mathbf{q}}$ is related to the generalized velocities of the bodies $\mathbf{u}$:

$$\dot{\mathbf{q}} = \mathbf{H}(\mathbf{q})\mathbf{u}. \tag{16}$$

where $\mathbf{H}(\mathbf{q})$ is the generalized kinematic map. The diagonal blocks $\mathbf{H}_{jj}$ are given by equation 1 and off diagonal blocks are zero.

Joint Constraints: Since joints are permanent contacts, if contact i is a joint (*i.e.,* $i \in \mathcal{B}$), then the vector function ${}^{b}\mathbf{C}_{in}(\mathbf{q},t) = \mathbf{0}$ for all time. Stacking the ${}^{b}\mathbf{C}_{in}$ functions for all $i \in \mathcal{B}$ into the vector ${}^{b}\mathbf{C}_{n}(\mathbf{q},t)$, yields the position-level constraint for all joints:

$${}^{b}\mathbf{C}_{n}(\mathbf{q},t) = \mathbf{0}. \tag{17}$$

From a physical perspective, these constraints are maintained by reaction forces ${}^{b}\mathbf{f}_{in}$ that are unconstrained. That is, generalized forces normal or anti-normal to the constraint surface in the system's configuration space can be generated. When viewing multibody dynamics from a variational perspective, these forces are Lagrange multipliers [Lan86].

Normal Contact Constraints: For the unilateral contacts, the scalar functions, ${}^{u}\mathbf{C}_{in}(\mathbf{q},t)$ for all $i \in \mathcal{U}$ must be non-negative. Stacking all the gap functions into the vector ${}^{u}\mathbf{C}_{n}(\mathbf{q},t)$ yields the following position-level non-penetration constraint:

$${}^{u}\mathbf{C}_{n}(\mathbf{q},t) \geq \mathbf{0}. \tag{18}$$

From a physical perspective, this constraint is maintained by the normal component of the contact force ${}^{u}\mathbf{f}_{in}$ between the bodies. Again, this force can be viewed as a Lagrange multiplier, but since the constraint is one-sided, so is the multiplier (*i.e.,* ${}^{u}\mathbf{f}_{in} \geq 0$). This means that constraint forces at unilateral contacts must be compressive or zero. Combining all ${}^{u}\mathbf{f}_{in}$ for all $i \in \mathcal{U}$ into the vector ${}^{u}\mathbf{f}_{n}$, we write all normal force constraints as:

$${}^{u}\mathbf{f}_{n} \geq \mathbf{0}. \tag{19}$$

There is one more aspect of unilateral contacts that must be modeled. If contact i is supporting a load (*i.e.,* ${}^{u}\mathbf{f}_{in} > 0$), then the contact must be maintained (*i.e.,* ${}^{u}\mathbf{C}_{in} = 0$). Conversely, if the contact breaks (*i.e.,* ${}^{u}\mathbf{C}_{in} > 0$), then the normal components (and hence the frictional components) of the contact force must be zero (*i.e.,* ${}^{u}\mathbf{f}_{in} = 0$). For each contact, at least one of ${}^{u}\mathbf{f}_{in}$ and ${}^{u}\mathbf{C}_{in}$ must be zero, (*i.e.,* ${}^{u}\mathbf{C}_{in}\ {}^{u}\mathbf{f}_{in} = 0$). These conditions are imposed at every contact simultaneously by an orthogonality constraint:

$${}^{u}\mathbf{C}_{n}(\mathbf{q},t) \;\cdot\; {}^{u}\mathbf{f}_{n} = 0 \tag{20}$$

where $\cdot$ denotes the vector dot product.

Friction Law: At contact i, the generalized friction force ${}^{\kappa}\mathbf{f}_{if}$ can act only in a subset of the unconstrained directions and must lie within a closed convex limit set $\mathcal{F}_i({}^{\kappa}\mathbf{f}_{in}, \mu_i)$. The

limit set must contain the origin, so that a zero friction force is possible. Also, typically, the limit set scales linearly with the normal component of the contact force, thus forming a cone of possible contact forces.

When contact i is rolling, the friction force may take on any value within the limit set. However, when the contact is sliding, the friction force must be the one within $\mathcal{F}_i({}^{\kappa}\mathbf{f}_{in}, \mu_i)$ that maximizes the power dissipation. Such models are said to satisfy the principle of maximum dissipation [Goy89]. At the velocity level, maximum dissipation can be expressed as follows:

$$ {}^{\kappa}\mathbf{f}_{if} \in \arg\max_{\mathbf{f}'_{if}} \left\{ -{}^{\kappa}\mathbf{v}_{if} \cdot \mathbf{f}'_{if} : \mathbf{f}'_{if} \in \mathcal{F}_i({}^{\kappa}\mathbf{f}_{in}, \mu_i) \right\} \tag{21} $$

where $\mathbf{f}'_{if}$ is an arbitrary vector in the set $\mathcal{F}_i({}^{\kappa}\mathbf{f}_{in}, \mu_i)$. Notice that when this set is strictly convex, then the friction force will be unique. For example, under the assumption of isotropic Coulomb friction at a unilateral contact, the limit set is the disc $\mu_i^2 {}^{u}\mathbf{f}_{in}^2 - {}^{u}\mathbf{f}_{it}^2 - {}^{u}\mathbf{f}_{io}^2 \geq 0$ and the unique friction force is the one directly opposite the relative sliding velocity, $({}^{u}\mathbf{v}_{it},\ {}^{u}\mathbf{v}_{io})$.

Finally, our instantaneous-time dynamic model is the system of differential algebraic inequalities (DAIs) composed of equations (13,16–21), where the sliding and rolling contact sets are defined as in equations (11) and (12) In the current form, the DAI is difficult to solve. However, as will be shown, it is possible to cast the model as a differential complementarity problem [CPS92b,TPSL97], then discretize the result to form simulation subproblems in the form of nonlinear or linear complementarity problems, allowing one to apply well-studied solution algorithms.

Complementarity Problems

The standard nonlinear complementarity problem (NCP) can be stated as follows:

Definition 1 Nonlinear Complementarity Problem: Given an unknown vector $\mathbf{x} \in \mathbb{R}^m$ and a known vector function $\mathbf{y}(\mathbf{x}) : \mathbb{R}^m \to \mathbb{R}^m$, determine $\mathbf{x}$ such that:

$$ \mathbf{0} \leq \mathbf{y}(\mathbf{x}) \perp \mathbf{x} \geq \mathbf{0}, \tag{22} $$

where $\perp$ implies orthogonality (*i.e.*, $\mathbf{y}(\mathbf{x}) \cdot \mathbf{x} = 0$).

The standard linear complementarity problem (LCP) is a special case in which the function $\mathbf{y}(\mathbf{x})$ is linear in $\mathbf{x}$:

Definition 2 Linear Complementarity Problem: Given an unknown vector $\mathbf{x} \in \mathbb{R}^m$, a known fixed matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$, and a known fixed vector $\mathbf{b} \in \mathbb{R}^m$, determine $\mathbf{x}$ such that:

$$ \mathbf{0} \leq \mathbf{A}\mathbf{x} + \mathbf{b} \perp \mathbf{x} \geq \mathbf{0}. \tag{23} $$

We adopt the shorthand notation, LCP($\mathbf{A}, \mathbf{b}$).

3.1.1. Complementarity Formulation of the Instantaneous-Time Model

To achieve model formulation as a properly posed complementarity problem, we must write all the conditions (13,16–21) in terms of a common small set of dependent variables. In the current formulation, the dependent variables are positions, velocities, and accelerations. However, by taking the appropriate number of time derivatives, all equations will be written in terms of accelerations, thus generating a model in which all dependent variables are forces and accelerations. This transformation will be carried out below in three steps. First, express the principle of maximum dissipation as a system of equations and inequalities in forces and accelerations, second, reformulate the Newton-Euler equation to expose the forces, and third, differentiate the distance functions twice with respect to time to expose the accelerations.

Reformulation of maximum dissipation The principle of maximum dissipation (21) can be replaced by an equivalent system of equations and inequalities by formulating it as an unconstrained optimization problem, and solving it in closed form. To do this, however, one must choose a specific form of $\mathcal{F}_i$. In this paper, we will demonstrate the solution process for isotropic Coulomb friction at a unilateral contact and apply the result to dry friction of constant maximum magnitude in a one-degree-of-freedom joint. The same procedure can be applied to other friction models, including Contensou [TP97, TTP01].

Closed-form solutions of optimization problems can sometimes be found by obtaining a system of equations corresponding to necessary and sufficient conditions for an optimal solution, then solving them. The most common approach is to augment the objective function with the constraints multiplied by Lagrange multipliers and then obtain the equations, known as the Karush-Kuhn-Tucker (KKT) equations, by partial differentiation. To be valid, the system must satisfy a regularity condition (also known as, "constraint qualification"). In the case of isotropic Coulomb friction, the system does not satisfy any of the possible regularity conditions at the point of the cone (where ${}^{u}\mathbf{f}_{in} = 0$), so the method fails.

Fortunately, the more general Fritz-John conditions [MF67] *do* satisfy a regularity condition everywhere on the cone. In the case of isotropic Coulomb friction, the augmented objective function (recall equation (21) is:

$$ -{}^{u}\beta_{i0}({}^{u}\mathbf{f}_{it}\,{}^{u}\mathbf{v}_{it} + {}^{u}\mathbf{f}_{io}\,{}^{u}\mathbf{v}_{io}) + {}^{u}\beta_i(\mu_i^2 {}^{u}\mathbf{f}_{in}^2 - {}^{u}\mathbf{f}_{it}^2 - {}^{u}\mathbf{f}_{io}^2), \tag{24} $$

where ${}^{u}\beta_i$ and ${}^{u}\beta_{i0}$ are Lagrange multipliers. To obtain a system of equations and inequalities equivalent to the maximum dissipation condition (21), one takes partial derivatives with respect to the unknown friction force components and Lagrange multipliers and then imposes the additional conditions of the Fritz-John method: ${}^{u}\beta_{i0} \geq 0$ and $({}^{u}\beta_{i0}, {}^{u}\beta_i) \neq (0,0)$. Following the derivation on pages 28-30 of [Ber09],

one arrives at the following system of constraints:

$$\left.\begin{aligned} \mu_i\, {}^u\mathbf{f}_{in}\, {}^u\nu_{it} + {}^u\mathbf{f}_{it}\, {}^u\beta_i = 0 \\ \mu_i\, {}^u\mathbf{f}_{in}\, {}^u\nu_{io} + {}^u\mathbf{f}_{io}\, {}^u\beta_i = 0 \\ {}^u\alpha_i = \mu_i^2\, {}^u\mathbf{f}_{in}^2 - {}^u\mathbf{f}_{it}^2 - {}^u\mathbf{f}_{io}^2 \geq 0 \\ 0 \leq {}^u\alpha_i \perp {}^u\beta_i \geq 0 \end{aligned}\right\} \quad \forall\, i \in \{\mathcal{U} \cap \mathcal{S}\}, \quad (25)$$

where ${}^u\alpha_i$ is a slack variable for the friction limit set. Note that ${}^u\beta_i = \| {}^u\nu_{if} \|$ at the optimal solution, and represents the magnitude of the slip velocity at contact i (*i.e.*, ${}^u\beta_i = \| {}^u\nu_{if} \| = \sqrt{{}^u\nu_{it}^2 + {}^u\nu_{io}^2}$). Note that this condition is *not* written in terms of accelerations, because at a sliding contact, the friction force $({}^u\mathbf{f}_{it},\, {}^u\mathbf{f}_{io})$ can be written in terms of the normal force and eliminated. For example, in the case of Coulomb friction, equations (5) and (6) are used. Tangential acceleration at a contact does not enter unless the contact point is rolling.

If contact i is a one-degree-of-freedom joint, we will assume that the maximum magnitude of the dry friction force is independent of the load in the other five component directions. Thus, the friction limit set for a bilateral joint $\mathcal{F}_i(\mu_i)$ will be:

$$\mathcal{F}_i({}^b\mathbf{f}_{if_{\max}}) = \left\{ {}^b\mathbf{f}_{if} : \left| {}^b\mathbf{f}_{if} \right| \leq {}^b\mathbf{f}_{if_{\max}} \right\}, \quad \forall\, i \in \{\mathcal{B} \cap \mathcal{S}\} \quad (26)$$

where $|\cdot|$ denotes the absolute value of a scalar and ${}^b\mathbf{f}_{if_{\max}}$ is the nonnegative maximum magnitude of the generalized friction force in joint i.

Notice that this joint friction model is a special case of the result obtained for Coulomb friction; fix ${}^u\mathbf{f}_{in}\mu_i$ to the value of ${}^b\mathbf{f}_{if_{\max}}$ and remove one of the friction directions, say the t-direction. The result is:

$$\left.\begin{aligned} {}^b\mathbf{f}_{io_{\max}}\, {}^b\nu_{io} + {}^b\mathbf{f}_{io}\, {}^b\beta_i = 0 \\ {}^b\alpha_i = {}^b\mathbf{f}_{io_{\max}}^2 - {}^b\mathbf{f}_{io}^2 \geq 0 \\ 0 \leq {}^b\alpha_i \perp {}^b\beta_i \geq 0 \end{aligned}\right\} \quad \forall\, i \in \mathcal{B}. \quad (27)$$

As before, ${}^b\beta_i = \| {}^b\nu_{if} \|$ at an optimal solution.

Contact Constraints in Terms of Accelerations Contact constraints (unilateral and bilateral) can be written in terms of accelerations through Taylor series expansion of constraint functions, ${}^\kappa C_{i\sigma}(q,t)$; $\kappa \in \{b,\, u\}$; $\sigma \in \{n,\, f\}$ Let $\tilde{q} = q + \Delta q$ and $\tilde{t} = t + \Delta t$ where Δq and Δt are small perturbations.

Then the Taylor expansion of one of these scalar displacement functions truncated to the quadratic terms is:

$$\begin{aligned} \widehat{{}^\kappa C}_{i\sigma}(\tilde{\mathbf{q}},\tilde{t}) &= {}^\kappa C_{i\sigma}(\mathbf{q},t) \\ &+ \frac{\partial {}^\kappa C_{i\sigma}}{\partial \mathbf{q}}\Delta\mathbf{q} + \frac{\partial {}^\kappa C_{i\sigma}}{\partial t}\Delta t \\ &+ \frac{1}{2}\left((\Delta\mathbf{q})^T \frac{\partial^2 {}^\kappa C_{i\sigma}}{\partial \mathbf{q}^2}\Delta\mathbf{q} + 2\frac{\partial^2 {}^\kappa C_{i\sigma}}{\partial \mathbf{q}\partial t}\Delta\mathbf{q}\Delta t + \frac{\partial^2 {}^\kappa C_{i\sigma}}{\partial t^2}\Delta t^2 \right) \end{aligned}$$

Notice that if contact exists at the current values of $\mathbf{q}$ and t, then the first term is zero. Dividing the linear terms by Δt and taking the limit as Δt (and $\Delta\mathbf{q}$) goes to zero, one obtains the relative velocity, ${}^\kappa\nu_{i\sigma}$ at the contact. Dividing the quadratic terms by $(\Delta t)^2$ and taking the limit yields the relative acceleration ${}^\kappa\mathbf{a}_{i\sigma}$:

$${}^\kappa\mathbf{a}_{i\sigma} = {}^\kappa\mathbf{J}_{i\sigma}\dot{\mathbf{u}} + {}^\kappa\mathbf{k}_{i\sigma}(\mathbf{q},\mathbf{u},t) \quad (28)$$

where

$${}^\kappa\mathbf{J}_{i\sigma} = \frac{\partial({}^\kappa\mathbf{C}_{i\sigma})}{\partial\mathbf{q}}\mathbf{H}$$

$${}^\kappa\mathbf{k}_{i\sigma}(\mathbf{q},\mathbf{u},t) = \frac{\partial({}^\kappa\mathbf{C}_{i\sigma})}{\partial\mathbf{q}}\frac{\partial\mathbf{H}}{\partial t}\mathbf{u} + \frac{\partial^2({}^\kappa\mathbf{C}_{i\sigma})}{\partial\mathbf{q}\partial t}\mathbf{H}\mathbf{u} + \frac{\partial^2({}^\kappa\mathbf{C}_{i\sigma})}{\partial t^2},$$

where recall that κ is either b or u and σ is either n or f.

Stacking all the quantities above for every unilateral and bilateral contact (as defined in equations (11) and (12)), one arrives at the definitions of ${}^\kappa\mathbf{a}_n$, ${}^\kappa\mathbf{J}_n$, and ${}^\kappa\mathbf{k}_n$, which allows us to express equations (17-20) in terms of accelerations as follows:

$${}^b\mathbf{a}_n = \mathbf{0}. \quad (30)$$

$$\mathbf{0} \leq {}^u\mathbf{f}_n \perp {}^u\mathbf{a}_n \geq \mathbf{0}. \quad (31)$$

The principle of maximum dissipation (21) must be considered further. When contact i is sliding, the solutions of conditions (25) and (27) produce the correct results (*i.e.,* the friction force obtains its maximum magnitude and directly opposes the sliding direction) and, we can use these conditions to eliminate ${}^\kappa\mathbf{f}_{if}$. Also as required, when a contact is rolling, these conditions allow the friction force to lie anywhere within the friction limit set. What these conditions do not provide is a mechanism for determining if a rolling contact will change to sliding. However, this problem is easily remedied by replacing the relative velocity variables in equation (21) with the analogous acceleration variables.

$$\left.\begin{aligned} \mu_i\, {}^u\mathbf{f}_{in}\, {}^u\mathbf{a}_{it} + {}^u\mathbf{f}_{it}\, {}^u\beta_i = 0 \\ \mu_i\, {}^u\mathbf{f}_{in}\, {}^u\mathbf{a}_{io} + {}^u\mathbf{f}_{io}\, {}^u\beta_i = 0 \\ {}^u\beta_i = \mu_i^2\, {}^u\mathbf{f}_{in}^2 - {}^u\mathbf{f}_{it}^2 - {}^u\mathbf{f}_{io}^2 \geq 0 \\ 0 \leq {}^u\beta_i \perp {}^u\beta_i \geq 0 \end{aligned}\right\} \quad \forall\, i \in \mathcal{U} \cap \mathcal{R} \quad (32)$$

where ${}^u\beta_i = \| {}^u\mathbf{a}_{if} \|$ at the optimal solution, and

$$\left.\begin{aligned} {}^b\mathbf{f}_{if_{\max}}\, {}^b\mathbf{a}_{if} + {}^b\mathbf{f}_{if}\, {}^b\beta_i = 0 \\ {}^b\beta_i = {}^b\mathbf{f}_{if_{\max}}^2 - {}^b\mathbf{f}_{if}^2 \geq 0 \\ 0 \leq {}^b\beta_i \perp {}^b\beta_i \geq 0 \end{aligned}\right\} \quad \forall\, i \in \mathcal{B} \cap \mathcal{R} \quad (33)$$

where ${}^b\mathbf{f}_{if} = \left| {}^b\mathbf{a}_{if} \right|$ at the optimal solution.

Exposing the Contact Forces in the Newton-Euler Equation Recall that the vector $\mathbf{g}(\mathbf{q},\mathbf{u},t)$ represents the resultant generalized forces acting on the bodies, and naturally generated gyroscopic forces. In order to complete the formulation as an NCP, $\mathbf{g}(\mathbf{q},\mathbf{u},t)$ is expressed as the sum of the normal

and friction forces at the unilateral and bilateral contacts and all other generalized forces. The Newton-Euler equation becomes:

$$\begin{aligned}\mathbf{M}(\mathbf{q})\dot{\mathbf{u}} = {}^{u}\mathbf{J}_n(\mathbf{q})^T {}^{u}\mathbf{f}_n + {}^{u}\mathbf{J}_f(\mathbf{q})^T {}^{u}\mathbf{f}_f \\ + {}^{b}\mathbf{J}_n(\mathbf{q})^T {}^{b}\mathbf{f}_n + {}^{b}\mathbf{J}_f(\mathbf{q})^T {}^{b}\mathbf{f}_f + \mathbf{g}_{\text{ext}}(\mathbf{q},\mathbf{u},t)\end{aligned} \tag{34}$$

where $\mathbf{g}_{\text{ext}}(\mathbf{q},\mathbf{u},t)$ is the resultant of all non-contact forces and moments applied to the bodies, ${}^{u}\mathbf{f}_f$ and ${}^{b}\mathbf{f}_f$ are formed by stacking the generalized friction vectors at the unilateral and bilateral contacts respectively, and the matrices ${}^{\kappa}\mathbf{J}_\sigma$ map contact forces into a common inertial frame.

3.1.2. A Differential NCP

The instantaneous dynamic model is now complete.

Definition 3 CP1: Equations (16,25,27,30–34) constitute a differential, nonlinear complementarity problem.

It is known that solutions to CP1 do not always exist (see [TPSL97]), however, if the principle of maximum dissipation is relaxed so that friction forces merely need to be dissipative, rather than maximally dissipative, then a solution always exists [PT96]. If one wanted to use this NCP in an integration scheme to simulate the motion of a multibody system, the PATH algorithm by Ferris and Munson is the most robust, general purpose NCP solver available [CPN11]. One should also note that this NCP can be converted into an approximate LCP by linearizing the friction cone constraint (see [TPSL97] for details). The LCP then can be solved by Lemke's algorithm [CPS92b], but the solution non-existence problem persists unless the maximum dissipation requirement is relaxed as just stated above.

3.1.3. A Nonlinear Discrete-Time Model

We recommend against applying a integration method to the instantaneous-time model due the possible non-existence of computable solutions (*i.e.,* a values of the unknown accelerations and constraint forces might not exist *OR* the solution might contain infinite values). However, this problem can be alleviated by applying discrete-time derivative approximations to the model over a time step Δt and then recasting the model as a time-stepping complementarity problem whose unknowns are impulses (time integrals of forces) and velocities (time integrals of accelerations). We demonstrate the process in this section.

Let Δt denote a positive step size and t_ℓ the current time, for which we have estimates of the configuration $\mathbf{q}^{(\ell)} = \mathbf{q}(t_\ell)$ and the generalized velocity $\mathbf{u}^{(\ell)} = \mathbf{u}(t_\ell)$ of the system. Our goal is to compute configurations $\mathbf{q}^{(\ell+1)} = \mathbf{q}(t_\ell + \Delta t)$ and velocities $\mathbf{u}^{(\ell+1)} = \mathbf{u}(t_\ell + \Delta t)$ that lie as close as possible to a solution of the differential NCP, CP1.

In the derivation, we will require estimates of the matrices $\mathbf{M}$, ${}^{\kappa}\mathbf{J}_\sigma$, and the vector $\mathbf{g}_{\text{ext}}$, which all vary with the system configuration $\mathbf{q}$. As will be seen, the simplest choice will be to use their values at $\mathbf{q}^{(\ell)}$, denoted by $\mathbf{M}^{(\ell)}$, ${}^{\kappa}\mathbf{J}_\sigma^{(\ell)}$, and $\mathbf{g}_{\text{ext}}^{(\ell)}$, because that will result in an explicit time-stepping method with each step requiring the solution of an NCP, whose only nonlinearity is the quadratic constraint of the friction cone. Approximating the cone constraint with a polyhedral cone will replace the quadratic constraints with linear ones, and thus will convert the NCP into an LCP, which is typically easier to solve.

In the following section, we discretize the five components of the instantaneous-time model derived above. To simplify our presentation, we choose the simple backward Euler approximation of the state derivatives, *i.e.,* $\dot{\mathbf{u}}(t_{\ell+1}) \approx (\mathbf{u}^{(\ell+1)} - \mathbf{u}^{(\ell)})/\Delta t$ and $\dot{\mathbf{q}}(t_{\ell+1}) \approx (\mathbf{q}^{(\ell+1)} - \mathbf{q}^{(\ell)})/\Delta t$.

Discrete-Time Newton-Euler Equations Applying the backward Euler approximation to the Newton-Euler equation (34) yields the equation below in which all quantities are evaluated at the end of the time step:

$$\begin{aligned}\mathbf{M}^{(\ell+1)}\left(\mathbf{u}^{(\ell+1)} - \mathbf{u}^{(\ell)}\right) = ({}^{u}\mathbf{J}_n^T)^{(\ell+1)} {}^{u}\mathbf{p}_n^{(\ell+1)} \\ + ({}^{u}\mathbf{J}_f^T)^{(\ell+1)} {}^{u}\mathbf{p}_f^{(\ell+1)} + ({}^{b}\mathbf{J}_n^T)^{(\ell+1)} {}^{b}\mathbf{p}_n^{(\ell+1)} \\ + ({}^{b}\mathbf{J}_f^T)^{(\ell+1)} {}^{b}\mathbf{p}_f^{(\ell+1)} + (\mathbf{p}_{\text{ext}})^{(\ell+1)},\end{aligned} \tag{35}$$

where the vectors are unknown generalized contact impulses defined as $\mathbf{p}^{(\ell+1)} = \Delta t \mathbf{f}^{(\ell+1)}$ and $\mathbf{p}_{\text{ext}} = \Delta t \mathbf{g}_{\text{ext}}$ is the impulse of the generalized forces applied to the bodies over the time step.

Discrete-Time Kinematic Map Applying the backward Euler approximation to the kinematic map (16) gives:

$$\mathbf{q}^{(\ell+1)} - \mathbf{q}^{(\ell)} \;=\; \Delta t \mathbf{H}^{(\ell+1)} \mathbf{u}^{(\ell+1)}, \tag{36}$$

which is nonlinear in the unknown system configuration $\mathbf{q}^{(\ell+1)}$. An important issue arises when solving this equation for $\mathbf{q}^{(\ell+1)}$. The "vector" $\mathbf{q}^{(\ell+1)}$, is *NOT* a vector; the orientation part of $\mathbf{q}$ lives in a curved space, not a vector space. For example, when orientation is represented by a unit quaternion, then quaternion elements of $\mathbf{q}^{(\ell)}$ and $\mathbf{q}^{(\ell+1)}$ must have unit length, but adding $\Delta t \mathbf{H}^{(\ell+1)} \mathbf{u}^{(\ell+1)}$ to $\mathbf{q}^{(\ell)}$ slightly increases the length. This problem can be solved simply by normalizing the quaternion elements of $\mathbf{q}^{(\ell+1)}$ after each time step.

Discrete-Time Contact Constraints The contact and joint constraints given in equations (17,18,20) can be incorporated into the discrete-time framework by replacing the configuration variables with their values at time $t_{(\ell+1)}$. An alternative is to expand the functions in a Taylor series and choose the point of truncation, to control the level of accuracy and nonlinearity. Denoting ${}^{\kappa}\mathbf{C}_\sigma\left(\mathbf{q}^{(\ell)}, t^{(\ell)}\right)$ by ${}^{\kappa}\mathbf{C}_\sigma^{(\ell)}$, a

Taylor series expansion is:

$$\widehat{{}^{\kappa}\mathbf{C}_{\sigma}^{(\ell+1)}} = {}^{\kappa}\mathbf{C}_{\sigma}^{(\ell)} + ({}^{\kappa}\mathbf{J}_{\sigma})^{(\ell)}\mathbf{u}^{(\ell+1)}\Delta t + \frac{\partial {}^{\kappa}\mathbf{C}_{\sigma}^{(\ell)}}{\partial t}\Delta t + H.O.T, \tag{37}$$

where the hat over the $\mathbf{C}$ in denotes an approximation and $H.O.T.$ denotes higher-order terms.

Discrete-Time Normal Contact Constraints Given that the discrete time Newton-Euler equations are written in terms of unknown impulses, equation (19) should also be converted to impulses. Since ${}^{u}\mathbf{p}_{n}^{(\ell+1)} = \Delta t\, {}^{u}\mathbf{f}_{n}^{(\ell+1)}$ and Δt is strictly positive, equation (19) becomes:

$${}^{u}\mathbf{p}_{n}^{(\ell+1)} \geq 0. \tag{38}$$

Last, for each unilateral contact, we must enforce complementarity between the gap function at the end of the time step and the normal impulse. Inserting the Taylor series approximation into equation (20) yields:

$$0 \leq {}^{u}\mathbf{p}_{n}^{(\ell+1)} \perp \widehat{{}^{u}\mathbf{C}_{n}^{(\ell+1)}} \geq 0. \tag{39}$$

Note that, in general, the right-hand inequality is nonlinear in the unknown $\mathbf{u}^{(\ell+1)}$. However, it can be made linear by truncating the Taylor series after the linear terms. It is also important to see that this relationship implies that the normal impulse ${}^{u}\mathbf{p}_{n}^{(\ell+1)}$ at the end of the time step can be nonzero only if the approximated distance function at the end of the time step is zero.

Discrete-Time Maximum Dissipation Principle To modify the maximum dissipation condition for use in time stepping, one integrates the force over a short time interval to obtain an impulse. If the direction of sliding changes little over the time step, then the friction law can be well approximated by simply replacing force variables with impulse variables. Thus, equation (21) becomes:

$$\begin{aligned} {}^{\kappa}\mathbf{p}_{if}^{(\ell+1)} \in \arg\max_{\mathbf{p}'_{if}} \Big\{ -\left({}^{\kappa}\mathbf{v}_{if}^{(\ell+1)}\right)^{T} \mathbf{p}'_{if} \;:\; \\ \mathbf{p}'_{if} \in \mathcal{F}_i({}^{\kappa}\mathbf{p}_{in}^{(\ell+1)}, \mu_i) \Big\} \end{aligned} \tag{40}$$

where $\mathbf{p}'_{if}$ is an arbitrary vector in the set $\mathcal{F}_i({}^{\kappa}\mathbf{p}_{in}^{(\ell+1)}, \mu_i)$. As it was the case during the formulation of the instantaneous model, we cannot complete the formulation of the discrete-time model without assuming a particular form of $\mathcal{F}_i$.

3.1.4. The Discrete-Time Model as an LCP

To develop a discrete-time model in the form of an LCP, all equations and inequalities in the model must be linear in the unknown configuration $\mathbf{q}^{(\ell+1)}$, velocity $\mathbf{u}^{(\ell+1)}$, and impulse $(\mathbf{p}_{\text{ext}})^{(\ell+1)}$. The discrete-time Newton Euler equation (35) appears to be linear in the unknown impulses and velocities, but the Jacobian matrices are functions of the unknown configuration and the external impulse $(\mathbf{p}_{\text{ext}})^{(\ell+1)}$ is generally a function of both $\mathbf{q}^{(\ell+1)}$ and $\mathbf{u}^{(\ell+1)}$. The standard way to obtain a linear equation is to evaluate the Jacobians and the external impulse at t_ℓ.

$$\begin{aligned} \mathbf{M}^{(\ell)}\left(\mathbf{u}^{(\ell+1)} - \mathbf{u}^{(\ell)}\right) = ({}^{u}\mathbf{J}_{n}^{T})^{(\ell)}\, {}^{u}\mathbf{p}_{n}^{(\ell+1)} \\ + ({}^{u}\mathbf{J}_{f}^{T})^{(\ell)}\, {}^{u}\mathbf{p}_{f}^{(\ell+1)} + ({}^{b}\mathbf{J}_{n}^{T})^{(\ell)}\, {}^{b}\mathbf{p}_{n}^{(\ell+1)} \\ + ({}^{b}\mathbf{J}_{f}^{T})^{(\ell)}\, {}^{b}\mathbf{p}_{f}^{(\ell+1)} + (\mathbf{p}_{\text{ext}})^{(\ell)}. \end{aligned} \tag{41}$$

Another option is to extrapolate these quantities forward in time using quantities known as time t_ℓ, as done in [PAGT05]. A drawback of this approach is that it requires computation of derivatives of the matrices.

Equation (36) is also nonlinear due to the dependence of $\mathbf{H}$ on $\mathbf{q}^{(\ell+1)}$. As above, evaluating it at t_ℓ yields a linear approximation:

$$\mathbf{q}^{(\ell+1)} - \mathbf{q}^{(\ell)} \;=\; \Delta t \mathbf{H}^{(\ell)}\mathbf{u}^{(\ell+1)}. \tag{42}$$

As mentioned earlier, the distance function constraints can be made linear by truncating the Taylor series after the linear terms. For the bilateral contacts, equation (37) becomes:

$${}^{b}\mathbf{C}_{n}^{(\ell)} + ({}^{b}\mathbf{J}_{n})^{(\ell)}\mathbf{u}^{(\ell+1)}\Delta t + \frac{\partial {}^{b}\mathbf{C}_{n}^{(\ell)}}{\partial t}\Delta t = 0. \tag{43}$$

Similarly, the normal complementarity condition (39) becomes:

$$0 \leq {}^{u}\mathbf{p}_{n}^{(\ell+1)} \perp {}^{u}\mathbf{C}_{n}^{(\ell)} + ({}^{u}\mathbf{J}_{n})^{(\ell)}\mathbf{u}^{(\ell+1)}\Delta t + \frac{\partial {}^{u}\mathbf{C}_{n}^{(\ell)}}{\partial t}\Delta t \geq 0. \tag{44}$$

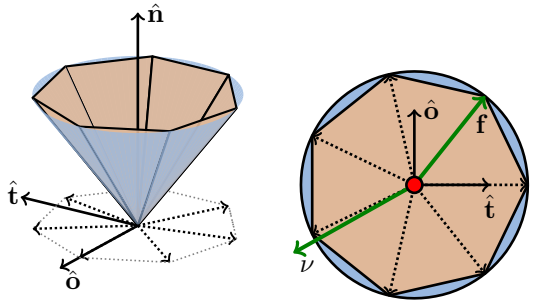
It only remains to linearize the friction model. To do so, however, requires a specific choice of friction limit set. Therefore, at this point, we will choose isotropic Coulomb friction and demonstrate the process of linearization for it, which is illustrated in Figure 7. The circular friction limit set is a circle of radius $\mu_i\, {}^{u}\mathbf{p}_{in}^{(\ell+1)}$ (shown in Figure 7(b)). This circle is approximated by a convex polygon whose vertices are defined by n_d the unit vectors $\hat{\mathbf{d}}_{ij}$ that positively span the friction plane.

To constrain the friction impulse at contact i, ${}^{u}\mathbf{p}_{if}^{(\ell+1)}$, to lie within the polygonal limit set, we employ nonnegative barycentric coordinates, ${}^{u}\alpha_{ij} \geq 0;\ j = \{1,...,n_d\}$. The interior and boundary of the linearized friction impulse limit set can be represented as follows:

$$\left.\begin{aligned} {}^{u}\mathbf{p}_{if}^{(\ell+1)} &= {}^{u}\mathbf{D}_i\, {}^{u}\alpha_i \\ \textstyle\sum_{j=1}^{n_d} {}^{u}\alpha_{ij} &\leq \mu_i\, {}^{u}\mathbf{p}_{in}^{(\ell+1)} \end{aligned}\right\} \quad \forall\, i \in \mathcal{U} \tag{45}$$

where ${}^{u}\mathbf{D}_i$ is the matrix whose j^{th} column is the unit vector $\hat{\mathbf{d}}_{ij}$ and ${}^{u}\alpha_i$ is the vector with j^{th} element given by ${}^{u}\alpha_{ij}$.

For the developments in the next few paragraphs, it is important to see that if ${}^{u}\alpha_{ij} = \mu_i\, {}^{u}\mathbf{p}_{in}^{(\ell+1)}$, then the friction impulse is simply $\mu_i\hat{\mathbf{d}}_{ij}$, which is a vertex of the polygon. If



(a) Friction Cone (b) Limit Set

Figure 7: *Example of friction cone linearization using seven friction direction vectors. Note the linearization side-effect - the friction force that maximizes dissipation is not exactly opposite the relative velocity at the contact point.*

${}^u\alpha_{ij} + {}^u\alpha_{ik} = \mu_i {}^u\mathbf{p}_{in}^{(\ell+1)}$, where $\hat{\mathbf{d}}_{ij}$ and $\hat{\mathbf{d}}_{ik}$ are adjacent direction vectors, then the friction impulse is on an edge of the polygon. Importantly, these are the *only* ways to represent friction impulses on the boundary of the limit set using barycentric coordinates. All other coordinate combinations define a friction impulse on the interior of the polygon.

It remains to enforce that at rolling contacts, the friction impulse must be within the limit set, but while sliding, it must maximize power dissipation, which requires the impulse to be on the boundary of the limit set. Let the non-negative slack variable ${}^u\beta_i$ be a scalar sliding indicator for contact i, where ${}^u\beta_i = 0$ implies rolling and ${}^u\beta_i > 0$ implies sliding. When ${}^u\beta_i = 0$, the friction impulse may be anywhere in the interior of the polygon or on its boundary, but when ${}^u\beta_i > 0$ it must be on the boundary.

These two requirements suggest a complementarity relationship between the representation (second equation in bracketed equations just above) and ${}^u\beta_i$:

$$0 \leq \left(\mu_i \mathbf{p}_{in}^{(\ell+1)} - {}^u\mathbf{e}_i^T {}^u\alpha_i\right) \perp {}^u\beta_i \geq 0 \;\; \forall\, i \in \mathcal{U}. \qquad (46)$$

where $\mathbf{e}_i$ is a vector length n_d with all elements equal to one. This condition ensures that the friction impulse is in the cone, but it does not enforce maximum dissipation. To achieve the latter, one must introduce another condition that allows only one or two consecutive barycentric coordinates to be nonzero. One way to accomplish this is by the introduction of another complementarity constraint that maps the relative velocity of the friction subspace $\nu_{if}^{(\ell+1)}$ onto the $\hat{\mathbf{d}}_{ij}$ vectors (this can be accomplished with ${}^u\mathbf{D}_i^T {}^u\mathbf{J}_{if}\mathbf{u}^{(\ell+1)}$) and identifies j such that $\hat{\mathbf{d}}_{ij}$ is most directly opposite to $\nu_{if}^{(\ell+1)}$. The following linear complementarity condition, in conjunction with condition (46), identifies the correct $\hat{\mathbf{d}}_{ij}$:

$$\mathbf{0} \leq \left({}^u\mathbf{D}_i^T {}^u\mathbf{J}_{if}\mathbf{u}^{(\ell+1)} + \mathbf{e}_i {}^u\beta_i\right) \perp {}^u\alpha_i \geq \mathbf{0} \;\forall\, i \in \mathcal{U}. \qquad (47)$$

Consider for a moment complementarity condition (47). If the contact is sliding ${}^u\beta_i > 0$, at least one element of ${}^u\alpha_i$ must be positive. The only way to have a positive element of ${}^u\alpha_i$ is to have at least one element of the expression on the left be zero, which can only happen when ${}^u\beta_i$ takes on its minimum value. Note that this minimum value can never be zero as long as the vectors $\hat{\mathbf{d}}_{ij}$; $i = 1,\ldots,n_d$ positively span the friction subspace, and furthermore, it approximates the slip speed, with the approximation converging to the exact slip speed as n_d goes to infinity.

One important side-effect of the above approximation of the principle of maximum dissipation is that a finite cone of relative velocities at contact i leads to exactly the same friction impulse. Even if the direction of sliding changes smoothly, the direction of the friction impulse jumps from one direction vector to the next.

Combining the tangential complementarity conditions for all unilateral contacts yields linear complementarity systems that replace equations (25) and (32) in the instantaneous-time model:

$$\begin{aligned} \mathbf{0} &\leq \left({}^u\mathbf{D}^T {}^u\mathbf{J}_f\mathbf{u}^{(\ell+1)} + {}^u\mathbf{E}{}^u\beta\right) \quad \perp \quad {}^u\alpha \geq \mathbf{0} \\ \mathbf{0} &\leq \left(\mathbf{U}\,{}^u\mathbf{p}_n^{(\ell+1)} - {}^u\mathbf{E}^T {}^u\alpha\right) \quad \perp \quad {}^u\beta \geq \mathbf{0} \end{aligned} \qquad (48)$$

where the column vectors ${}^u\alpha$ and ${}^u\beta$ are formed by stacking the vectors ${}^u\alpha_i$ and scalars ${}^u\beta_i$, ${}^u\mathbf{D}^T$ is formed by stacking the matrices ${}^u\mathbf{D}_i^T$, ${}^u\mathbf{E}$ is a block diagonal matrix with nonzero blocks given by ${}^u\mathbf{e}_i$, and $\mathbf{U}$ is the diagonal matrix with element (i,i) equal to μ_i.

If all joints in the system are one-degree-of-freedom joints, equations (27) and (33) of the instantaneous-time model are replaced with the following:

$$\begin{aligned} \mathbf{0} &\leq \left({}^b\mathbf{D}^T {}^b\mathbf{J}_f\mathbf{u}^{(\ell+1)} + {}^b\mathbf{E}{}^b\beta\right) \quad \perp \quad {}^b\alpha \geq \mathbf{0} \\ \mathbf{0} &\leq \left({}^b\mathbf{p}_{f\max} - {}^b\mathbf{E}^T {}^b\alpha\right) \quad \perp \quad {}^b\beta \geq \mathbf{0} \end{aligned} \qquad (49)$$

where the column vectors ${}^b\alpha$ and ${}^b\beta$ are formed by stacking the vectors ${}^b\alpha_i$ and scalars ${}^b\beta_i$, ${}^b\mathbf{D}^T$ is formed by stacking the matrices ${}^b\mathbf{D}_i^T$, ${}^b\mathbf{E}$ is a block diagonal matrix with nonzero blocks given by ${}^b\mathbf{e}_i$, and ${}^b\mathbf{p}_{f\max}$ is $\Delta t\ {}^b\mathbf{f}_{f\max}$.

3.1.5. A Time-Stepping LCP

Equations (41–44,48,49) constitute a discrete-time model in the form of a mixed LCP, that is "mixed," because it contains equations that cannot be directly put into the form of a linear complementarity condition: the Newton-Euler equation (41), the kinematic map (42), and the normal joint constraint (43). Notice, however, that the only place in the mixed LCP in which the unknown generalized position $\mathbf{q}^{(\ell+1)}$ appears is the kinematic map equation (42). Therefore, the mixed LCP can be decoupled into an LCP with unknown generalized velocities and impulses and the kinematic map

equation with unknown generalized positions. The decoupled systems can be solved in two steps: solve the smaller mixed LCP (LCP1 defined below), then use the solution to LCP1 in the kinematic map to update the generalized positions.

Definition 4 LCP1: A mixed LCP with unknown velocities and impulses is constituted by equations (41,43,44,48,49).

It is known that solutions always exist to LCP1 if the terms ${}^u\mathbf{C}_n^{(\ell)}$, $\frac{\partial {}^u\mathbf{C}_n^{(\ell)}}{\partial t}\Delta t$, and $\frac{\partial {}^b\mathbf{C}_n^{(\ell)}}{\partial t}\Delta t$ from equations (43) and (44) are removed (see [AP97a]).

The mixed LCP1 can be solved in its current form by the PATH algorithm [FM99] or it can first be reformulated as a standard LCP and then solved. Since $\mathbf{M}$ is symmetric and positive definite, and under the assumption that the null space of ${}^b\mathbf{J}_n^{(\ell)}$ is trivial (which is usually true if there are no kinematic loops in mechanisms), then one can solve for both $\mathbf{u}^{(\ell+1)}$ and ${}^b\mathbf{p}_n^{(\ell+1)}$ with equations (41) and (43) and substitute the results into equations (39,48,49). Before substitution these equations :

$$0 \le \begin{bmatrix} {}^u\mathbf{J}_n^{(\ell)}\mathbf{u}^{(\ell+1)} + \frac{{}^u\mathbf{C}_n^{(\ell)}}{\Delta t} + \frac{\partial {}^u\mathbf{C}_n^{(\ell)}}{\partial t} \\ {}^u\mathbf{D}^T {}^u\mathbf{J}_f\mathbf{u}^{(\ell+1)} + {}^u\mathbf{E}\,{}^u\beta \\ {}^b\mathbf{D}^T {}^b\mathbf{J}_f\mathbf{u}^{(\ell+1)} + {}^b\mathbf{E}\,{}^b\beta \\ \mathbf{U}\,{}^u\mathbf{p}_n^{(\ell+1)} - {}^u\mathbf{E}^T {}^u\alpha \\ {}^b\mathbf{p}_{f\max} - {}^b\mathbf{E}^T {}^b\alpha \end{bmatrix} \perp \begin{bmatrix} {}^u\mathbf{p}^{(\ell+1)} \\ {}^u\alpha \\ {}^b\alpha \\ {}^u\beta \\ {}^b\beta \end{bmatrix} \ge 0. \quad (50)$$

For the remainder of section 3.1.5, we will drop the superscript $^{(\ell)}$ on all matrices and constraints, since they are evaluated at time t_ℓ. Continuing with the derivation of LCP1, we cast equations (41) and (43) into matrix form:

$$\begin{bmatrix} \mathbf{M} & -{}^b\mathbf{J}_n^T \\ -{}^b\mathbf{J}_n & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(\ell+1)} \\ {}^b\mathbf{p}_n^{(\ell+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \quad (51)$$

where

$$\mathbf{x}_1 = {}^u\mathbf{J}_n^T {}^u\mathbf{p}_n^{(\ell+1)} + {}^b\mathbf{J}_f^T {}^b\mathbf{D}\,{}^b\alpha + {}^u\mathbf{J}_f^T {}^u\mathbf{D}\,{}^u\alpha + \mathbf{M}\mathbf{u}^{(\ell)} + \mathbf{p}_{\text{ext}}$$

$$\mathbf{x}_2 = \frac{{}^b\mathbf{C}_n}{\Delta t} + \frac{\partial {}^b\mathbf{C}_n}{\partial t}.$$

Inverting the matrix on the left side of equation (51) yields:

$$\begin{bmatrix} \mathbf{u}^{(\ell+1)} \\ {}^b\mathbf{p}_n^{(\ell+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{12}^T & \mathbf{S}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}. \quad (52)$$

Letting $\mathbf{B} = -{}^b\mathbf{J}_n\mathbf{M}^{-1}{}^b\mathbf{J}_n^T$, then $\mathbf{S}_{11}$, $\mathbf{S}_{12}$, and $\mathbf{S}_{22}$ are defined as follows:

$$\mathbf{S}_{11} = \mathbf{M}^{-1} + \mathbf{M}^{-1}\,{}^b\mathbf{J}_n^T\,\mathbf{B}^{-1}\,{}^b\mathbf{J}_n\,\mathbf{M}^{-1} \quad (53a)$$

$$\mathbf{S}_{12} = \mathbf{B}^{-1}\,{}^b\mathbf{J}_n\mathbf{M}^{-1} \quad (53b)$$

$$\mathbf{S}_{22} = \mathbf{B}^{-1}. \quad (53c)$$

Substituting back into inequalities (50) and using the shorthand ${}^\kappa\mathbf{J}_D = {}^\kappa\mathbf{J}_f\,{}^\kappa\mathbf{D}^T$ yields a standard LCP($\mathbf{A}$, $\mathbf{b}$) with $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{x}$ given as follows:

$$\mathbf{b} = \begin{bmatrix} {}^u\mathbf{J}_n\mathbf{r} + \frac{{}^u\mathbf{C}_n}{\Delta t} + \frac{\partial {}^u\mathbf{C}_n}{\partial t} \\ {}^u\mathbf{J}_D\mathbf{r} \\ {}^b\mathbf{J}_D\mathbf{r} \\ \mathbf{0} \\ {}^b\mathbf{p}_{f\max} \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} {}^u\mathbf{p}_n \\ {}^u\alpha \\ {}^b\alpha \\ {}^u\beta \\ {}^b\beta \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} {}^u\mathbf{J}_n\mathbf{S}_{11}{}^u\mathbf{J}_n^T & {}^u\mathbf{J}_n\mathbf{S}_{11}{}^u\mathbf{J}_D^T & {}^u\mathbf{J}_n\mathbf{S}_{11}{}^b\mathbf{J}_D^T & \mathbf{0} & \mathbf{0} \\ {}^u\mathbf{J}_D\mathbf{S}_{11}{}^u\mathbf{J}_n^T & {}^u\mathbf{J}_D\mathbf{S}_{11}{}^u\mathbf{J}_D^T & {}^u\mathbf{J}_D\mathbf{S}_{11}{}^b\mathbf{J}_D^T & {}^u\mathbf{E} & \mathbf{0} \\ {}^b\mathbf{J}_D\mathbf{S}_{11}{}^u\mathbf{J}_n^T & {}^b\mathbf{J}_D\mathbf{S}_{11}{}^u\mathbf{J}_D^T & {}^b\mathbf{J}_D\mathbf{S}_{11}{}^b\mathbf{J}_D^T & \mathbf{0} & {}^b\mathbf{E} \\ \mathbf{U} & -{}^u\mathbf{E}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -{}^b\mathbf{E}^T & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

where

$$\mathbf{r} = \mathbf{S}_{11}\left(\mathbf{M}\mathbf{u}^{(\ell)} + \mathbf{p}_{\text{ext}}\right) + \mathbf{S}_{12}\left(\frac{{}^b\mathbf{C}_n}{\Delta t} + \frac{\partial {}^b\mathbf{C}_n}{\partial t}\right). \quad (54)$$

Note that it is known that when there are no joints (*i.e.,* rows three and five are removed from $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{x}$ and columns three and five are removed from $\mathbf{A}$) and the sum $\frac{{}^u\mathbf{C}_n}{\Delta t} + \frac{\partial {}^u\mathbf{C}_n}{\partial t}$ is nonnegative, then a solution always exists.

3.2. Modeling of Articulated Bodies and Jointed Mechanics

An articulated-body is a system of rigid bodies connected by joints (see figure 8). Each joint defines a holonomic constraint which must be resolved during the simulation. A holonomic constraint reduces the number of degrees of freedom of the system permanently (see section 2.1).

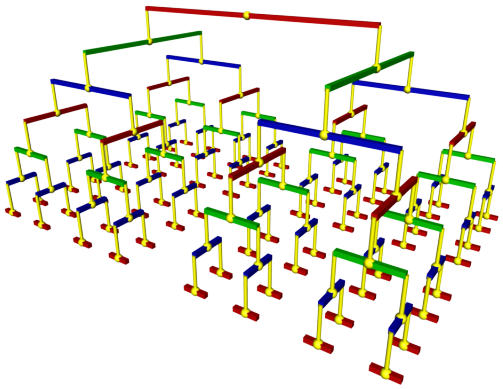


Figure 8: *This articulated body is a tree of rigid bodies which are connected by spherical joints.*

For the simulation of articulated bodies there exist two different formulations: the reduced (or generalized) coordinate formulation and the maximal coordinate formulation. The first one models the holonomic constraints by using a

reduced set of coordinates to describe the state of the system. An articulated-body has as many degrees of freedom as its number of independent coordinates. In contrast to that the maximal coordinate formulation uses the original coordinates of the rigid bodies and introduces additional forces or impulses in order to maintain the constraints.

3.2.1. Maximal coordinate formulation

The maximal coordinate formulation is well-known in the area of computer graphics. One of the first studies about multi-body simulation using this formulation was [BB88] which describes physically-based modeling with constraints. In the following, we will introduce different methods for this formulation.

Penalty force method The penalty force method can handle holonomic and nonholonomic equality constraints. These constraints are given in form of implicit functions in the form $\mathbf{C} = 0$. In a simulation step we can determine the violation of a constraint by evaluating its implicit function for the current state of the system. The result is zero if the constraint is fulfilled. Otherwise we add a force to the system in order to reduce the violation. For a holonomic equality constraint $\mathbf{C}(\mathbf{q},t) = 0$ this force can be computed by the following equation [dJB94]:

$$\mathbf{F}_{\text{penalty}} = -\alpha \mathbf{J}^T \left(\Omega^2 \mathbf{C} + 2\Omega\mu\dot{\mathbf{C}} + \ddot{\mathbf{C}}\right),$$

where $\mathbf{J}$ is the Jacobi matrix of the constraint $\mathbf{C}$. The values α, Ω and μ are constant parameters which are used to control the magnitude of the force. Multiplying the matrix $\mathbf{J}^T$ projects the force into the space of the constraint. The derivatives of the constraint $\dot{\mathbf{C}}$ and $\ddot{\mathbf{C}}$ are used in order to increase the stability. The resulting force is equivalent to the force of a damped spring with the spring constant α, natural frequency Ω and the damping ratio μ.

The force for a nonholonomic constraint is determined by

$$\mathbf{F}_{\text{penalty}} = -\alpha \left(\frac{\partial \mathbf{C}}{\partial \mathbf{u}}\right)^T (\mu\mathbf{C} + \dot{\mathbf{C}}).$$

For the simulation the penalty forces are added as external forces to the equation of motion. The penalty force method is easy to implement and very fast since the computation of penalty forces is very simple. The disadvantage of the method is that constraints can only be fulfilled approximately. An accurate solution is only possible for a very large value of α which leads to stiff differential equations.

Lagrange multipliers In contrast to the penalty force method the Lagrange multiplier method computes forces in order to prevent a violation of constraints. This method can simulate systems with holonomic and nonholonomic equality constraints. These constraints are transformed in the general constraint form

$$\mathbf{J}(\mathbf{q},\mathbf{u},t)\,\dot{\mathbf{u}} + \mathbf{k}(\mathbf{q},\mathbf{u},t) = \mathbf{0}. \tag{55}$$

In an n-dimensional system with an m-dimensional constraint the matrix $\mathbf{J}$ has the dimension $m \times n$ and the vector $\mathbf{k}$ the dimension m. A holonomic constraint is transformed in the general form by differentiating the constraint function $\mathbf{C}$ twice with respect to time. A nonholonomic equality constraint has just to be differentiated once. For a holonomic constraint $\mathbf{C}(\mathbf{q},t) = 0$ we get (cf. Equation 28)

$$\mathbf{J} = \frac{\partial \mathbf{C}}{\partial \mathbf{q}}\mathbf{H}, \quad \mathbf{k} = \frac{\partial \mathbf{C}}{\partial \mathbf{q}}\frac{\partial \mathbf{H}}{\partial t}\mathbf{u} + \frac{\partial^2 \mathbf{C}}{\partial \mathbf{q}\partial t}\mathbf{H}\mathbf{u} + \frac{\partial^2 \mathbf{C}}{\partial^2 t}.$$

A nonholonomic constraint $\mathbf{C}(\mathbf{q},\mathbf{u},t) = 0$ results in

$$\mathbf{J} = \frac{\partial \mathbf{C}}{\partial \mathbf{u}}, \quad \mathbf{k} = \frac{\partial \mathbf{C}}{\partial \mathbf{q}}\mathbf{H}\mathbf{u} + \frac{\partial \mathbf{C}}{\partial t}.$$

Most commonly in computer graphics the Lagrange multipliers λ are computed as follows. For each rigid body the mass matrix is defined by equation 14. The mass matrix for a particle with mass m is just the upper left block $m\mathbf{1}_{3\times 3}$. Constraints are simulated by additional forces $\mathbf{F}_c$ which are added to the equation of motion

$$\dot{\mathbf{u}} = \mathbf{M}^{-1}\left(\mathbf{F}_{\text{ext}} + \mathbf{F}_c\right) \tag{56}$$

where $\mathbf{M}$ is the mass matrix of the system which contains the mass matrices $\mathbf{M}_j$ of all bodies on the diagonal.

Substituting equation (56) into the general constraint (55), we obtain

$$\mathbf{J}\mathbf{M}^{-1}\mathbf{F}_c = -\mathbf{J}\mathbf{M}^{-1}\mathbf{F}_{\text{ext}} - \mathbf{k}.$$

Regarding D'Alembert's principle [GPS02], it follows that

$$\mathbf{F}_c = \mathbf{J}^T\lambda. \tag{57}$$

Hence, the constraint forces always act in the constrained directions of system. Such forces do not influence the motion of the $n-m$ degrees of freedom of an articulated body. Finally, we get a system of linear equations for the Lagrange multipliers

$$\underbrace{\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T}_{\mathbf{A}}\lambda = \underbrace{-\mathbf{J}\mathbf{M}^{-1}\mathbf{F}_{\text{ext}} - \mathbf{k}}_{\mathbf{b}}. \tag{58}$$

The matrix $\mathbf{A}$ is positive definite if there are no conflicting or redundant constraints. Furthermore, the matrix is sparse for the most models since it reflects the structure of the articulated body. After solving for λ the constraint forces are determined by equation 57.

One of the most well-known Lagrange multiplier methods in computer graphics is the one of David Baraff [Bar96]. This method allows the simulation of articulated bodies without closed loops in linear time. Only constraints that act between a pair of bodies are supported.

For a linear time computation the system of linear equations 58 is transformed in the following form

$$\underbrace{\begin{pmatrix}\mathbf{M} & -\mathbf{J}^T\\ -\mathbf{J} & \mathbf{0}\end{pmatrix}}_{\mathbf{K}}\begin{pmatrix}\mathbf{y}\\ \lambda\end{pmatrix} = \begin{pmatrix}\mathbf{0}\\ -\mathbf{b}\end{pmatrix}.$$

Matrix $\mathbf{K}$ is known as the KKT-matrix [NW99]. The matrix $\mathbf{A}$ is smaller than $\mathbf{K}$ and positive definite if $\mathbf{J}$ has full rank while $\mathbf{K}$ is not. $\mathbf{A}$ has a row and column for each constraint while $\mathbf{K}$ has a row and column for each degree of freedom and each constraint. The advantage of the new formulation is that $\mathbf{K}$ is always sparse and symmetric.

The next step is to create an undirected graph for $\mathbf{K}$ with a node for each block of the matrix and an edge between the nodes $i \neq j$ for each $\mathbf{K}_{ij} \neq \mathbf{0}$. This graph is acyclic since the model has no loops. By a depth search in this graph the matrix is reordered so that the row index that corresponds to a node in the graph is greater than the one of its children. Afterwards a $\mathbf{LDL}^T$ decomposition is performed. Due to the reordered matrix structure the decomposition introduces no new nonzero elements, can be stored in linear space and performed in linear time. The system of linear equations for the Lagrange multipliers can also be solved in linear time. Finally, the velocities and positions are determined by numerical integration.

The Lagrange multiplier method computes constraint forces in order to prevent a violation of constraints due to external forces. If the constraints are violated in a different way, e.g. by errors that occur during numerical integration, the method cannot correct this. These errors sum up over the simulation and the method is not able to prevent joints from breaking. The problem is that a holonomic constraint is not regarded directly. It is just demanded that its second derivative is zero. An error term of the form $\mathbf{k}_1 t + \mathbf{k}_2$ where $\mathbf{k}_1$ and $\mathbf{k}_2$ are two arbitrary constant vectors cannot be corrected in this way since

$$\frac{d^2}{d^2t}\mathbf{C}(\mathbf{q},t) = \frac{d^2}{d^2t}\left(\mathbf{C}(\mathbf{q},t) + \mathbf{k}_1 t + \mathbf{k}_2\right).$$

Therefore, joints will break due to numerical errors. An analogous problem exists for nonholonomic equality constraints. Therefore, an additional stabilization is required for the simulation with Lagrange multipliers.

The method of Baumgarte is often used for stabilization [Bau72]. This method replaces the equation $\ddot{\mathbf{C}} = \mathbf{0}$ of a holonomic constraint by

$$\ddot{\mathbf{C}} + 2\alpha\dot{\mathbf{C}} + \beta^2\mathbf{C} = \mathbf{0}$$

where α and β are constant parameters. In this way position and velocity errors are regarded in the simulation. The equation $\dot{\mathbf{C}} = \mathbf{0}$ of a nonholonomic equality constraint is replaced by

$$\dot{\mathbf{C}} + \gamma\mathbf{C} = \mathbf{0}$$

where the parameter γ defines the weight of the velocity error in the multiplier computation.

The stabilization terms can be added to the general form of a constraint (equation (55)). Alternatively, the terms can be taken into account by adding additional forces to the equation of motion [WW90]. The determination of suitable parameters for the stabilization is not easy. Ascher et al. discuss the problems of finding suitable parameters and propose an enhanced stabilization method [ACPR95].

Impulse-based simulation The impulse-based method [BFS05, BS06, Ben07, WTF06] is similar to the Lagrange multiplier method. The main difference between these methods is that the impulse-based approach determines impulses to perform a simulation with constraints by using a preview while the Lagrange multiplier method computes additional forces just regarding the current state.

Witkin et al. introduced a Lagrange multiplier method based on a constraint formulation with connectors [WGW90]. A connector is e.g. a point or vector in local coordinates of a body which is used to define a constraint. This allows to formulate generic constraints without knowledge about the object itself. For example, if two points are attached together, the corresponding constraint is $C(\mathbf{P}_1, \mathbf{P}_2) = \|\mathbf{P}_1 - \mathbf{P}_2\| = 0$ where the points $\mathbf{P}_1$ and $\mathbf{P}_2$ are connectors. The connector concept also solves the problem that the generalized position $\mathbf{q}$ and velocity $\mathbf{u}$ of a body commonly have not the same length (see Section 2.1).

In the following, constraints are defined by connectors $\mathbf{P}_i(\mathbf{q},t)$ fixed to the bodies:

$$\mathbf{C}(\mathbf{P}_1, \ldots, \mathbf{P}_m, t) = \mathbf{0}.$$

In this way the translational and rotational degrees of freedom can be constrained. By differentiating the constraint function $\mathbf{C}$ with respect to time we get a general constraint form for velocities $\mathbf{J}\mathbf{u} + \mathbf{k} = \mathbf{0}$ which is analogous to the one of equation 55. The matrix $\mathbf{J}$ and the vector $\mathbf{k}$ are determined by

$$\mathbf{J}_{ij} = \sum_k \frac{\partial \mathbf{C}_i}{\partial \mathbf{P}_k}\frac{\partial \mathbf{P}_k}{\partial \mathbf{q}_j}\mathbf{H}_j, \qquad \mathbf{k}_i = \frac{\partial \mathbf{C}_i}{\partial t} + \sum_k \frac{\partial \mathbf{P}_k}{\partial t}.$$

Now a system of linear equations for the impulses could be created which is analogous to the one of equation 58. However, the impulse-based method uses a different right hand side $\mathbf{b}$ for the system in order to solve the stabilization problem of the Lagrange multiplier method. The vector $\mathbf{b}$ is determined by a prediction of the joint state. This idea was first introduced by Bender et al. [BFS05] and later also used by Weinstein et al. [WTF06].

Figure 9 shows the preview of a ball joint with the holonomic constraint $\mathbf{C} = \mathbf{a} - \mathbf{b} = \mathbf{0}$. For the preview we assume that both rigid bodies are unconstrained. Then, we can easily compute the positions of the connector points $\mathbf{a}(t+h)$ and $\mathbf{b}(t+h)$ after a time step of size h by integration. The preview of a vector $\mathbf{r}$ fixed to a body is obtained by solving the differential equation

$$\dot{\mathbf{r}} = \boldsymbol{\omega} \times \mathbf{r}. \tag{59}$$

The predicted position of a point $\mathbf{a}$ is obtained by first solving equation 59 for the vector $\mathbf{r}(t) = \mathbf{a}(t) - \mathbf{x}(t)$ from the

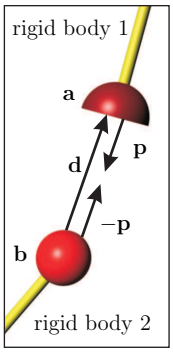


Figure 9: *Preview of a ball joint. The points* **a** *and* **b** *have different positions which must be corrected by a pair of impulses* **p** *and* $-\mathbf{p}$.

center of mass of the body to the point to get $\mathbf{r}(t+h)$. Then, we solve the equation of motion for the center of mass and determine the new position as $\mathbf{a}(t+h) = \mathbf{r}(t+h) + \mathbf{x}(t+h)$. For the predicted state we evaluate the constraint function and get the distance vector $\mathbf{d}(t+h) = \mathbf{a}(t+h) - \mathbf{b}(t+h)$. This vector shows us the violation which would occur without additional impulses in the system. Now we want to compute a pair of impulses $\mathbf{p}$ and $-\mathbf{p}$ for time t to prevent the violation. These impulses must cause a velocity change of the connectors so that the constraint $\mathbf{d}(t+h) = \mathbf{0}$ will be fulfilled.

The required impulses for a constraint can be determined by solving a nonlinear equation. Weinstein et al. use Newton iteration for the solution [WTF06]. In a system with multiple constraints there exist dependencies between the constraints if they have a common body. These dependencies are handled in an iterative way by Weinstein et al. In contrast to that Bender et al. linearize the equation by an approximation of the required velocity change. If the connectors have a linear relative motion, the required velocity change would be $\Delta\mathbf{v} = \mathbf{d}(t+h)/h$. Bender et al. use this value as an approximation for the nonlinear case which leads commonly to small errors [BS06]. These errors are eliminated by solving the following system for the impulses $\mathbf{p}$ iteratively

$$\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\mathbf{p} = \Delta\mathbf{v}$$

where $\Delta\mathbf{v}$ is the vector containing the velocity changes for all constraints.

Numerical comparisons and of the impulse-based approach with other simulation methods can be found in [**?**, **?**]. Bender showed in [Ben07] that the system of linear equations can be solved in linear time for articulated bodies without loops. In [**?**] the impulse-based method is extended by inequality constraints in order to simulate collisions and resting contacts. Bayer et al. [**?**] presented different optimizations for the impulse-based approach which increase the performance.

3.2.2. Reduced coordinate formulation

If we have a system of rigid bodies with m degrees of freedom and remove c of them by constraints, only the remaining $n = m - c$ degrees of freedom have to be simulated. The reduced coordinate formulation uses a parameterization for the m maximal coordinates in terms of n independent coordinates q_i which are called reduced or generalized coordinates. The m maximal coordinates of the system can be written as function of the reduced coordinates

$$x_i = x_i(q_1, \ldots, q_n), \quad i \in [1, \ldots, m].$$

Figure 10 shows an example for a reduced coordinate formulation. A particle with position $\mathbf{x} \in \mathbb{R}^2$ rotates around the

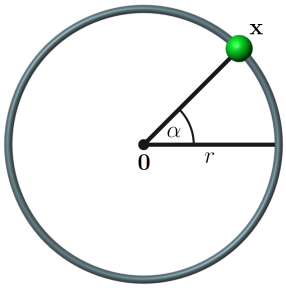


Figure 10: *The only degree of freedom of a particle moving on a circular path can be described by the angle* α.

origin on a circular path with radius r. This particle has only one degree of freedom which can be described by the angle α. The position in maximal coordinates is determined by

$$\mathbf{x}(\alpha) = r\begin{pmatrix}\cos\alpha\\ \sin\alpha\end{pmatrix}. \tag{60}$$

Equation 60 defines all valid positions $\mathbf{x}$ for the particle.

In order to perform a simulation with reduced coordinates we must first find a parameterization for our model. Then, we form the equations of motion with respect to the reduced coordinates. The resulting system of differential equations can be solved by numerical methods.

The equations of motion can be obtained by using the Lagrange formulation [GPS02, Fea07]. We require the Lagrangian function $L = T - V$ where T and V are the total kinetic and potential energy respectively. This function describes the difference of the total kinetic and potential energy of the system which should be conserved. By the Euler-Lagrange equation

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = 0, \quad i = 1, \ldots, n$$

we get a system of differential equations for the motion of the bodies which can be solved numerically. Methods based on Lagrange formulation have a complexity of $O(n^4)$.

An overview over more efficient reduced coordinate

methods can be found in [FO00]. One of them is the well-known articulated-body algorithm (ABA) of Featherstone with a complexity of $O(n)$ for articulated bodies with tree-structure [Fea87]. This algorithm works in two phases. In the first phase the kinematic parameters are determined considering external forces. The parameters of a body in the tree only depend on the parameters of its parent. Therefore, the kinematic parameters are computed in one traversal of the tree. In the second phase the tree is traversed in reverse. During this traversal the internal forces are determined for each body which just depend on the children of the body. A detailed description of the ABA of Featherstone can be found in [Mir96b] and [Fea07].

The method of Featherstone is used in different areas of computer graphics. One application area is the simulation of rag-dolls which have a tree-structure. These are used for example for improved motion synthesis techniques which combine motion capture data with physical simulation [MZS09]. There are also other application areas like the simulation of strands [Had06] or in games [Kok04]. Redon et al. [RGL05] presented an adaptive variant of Featherstone's method in order to improve the performance. This approach allows to reduce the numbers of degrees of freedom (at the cost of accuracy) while it automatically determines the best set of active joints.

The method of Featherstone works for models without kinematic loops. Models with loops cause problems which are discussed in detail by Wittenburg [Wit77]. An example is shown in Figure 11. The model consists of one static rigid body and five dynamic bodies where each has six degrees of freedom. Therefore, the free system has 30 degrees of freedom. If we create hinge joints between the bodies as shown in the figure, these degrees of freedom are reduced. Each of the six hinge joints eliminates five degrees of freedom in a loop-free model. Hence, we could expect that the model has no degrees of freedom left. But in fact it has still one.

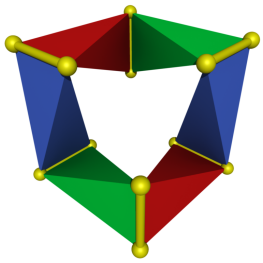


Figure 11: *Closed kinematic loop with one degree of freedom.*

The degrees of freedom of a closed-loop model can vary and forces in such a model can be indeterminate when the system is overconstrained. Therefore, these models needs some special treatment. A common approach to handle closed loops is to remove joints from the articulated body until we have a tree structure [FO00]. This is done by extracting a spanning tree from the connectivity graph. Now a simulation step is performed for the spanning tree and additional forces are added to mimic the effects of the kinematic loops. Loop handling is explained in detail in [Fea07].

4. The Numerical Solution Methods

Once discrete models have been obtained we must apply numerical methods to compute solutions. We start with how to integrate the motion of free moving rigid bodies such as bodies in ballistic motion without any collisions or contact. Subsequently in Sections 4.2- 4.4 we cover numerical methods for computing solutions of the discrete LCP contact model from Section 3.1.

4.1. Time Integration of Free Motion

For the simulation of constrained rigid bodies we extend the equation of motion by additional forces or impulses. Alternatively, we can formulate these equations in terms of reduced coordinates. In both cases we have to perform an integration step to obtain the dynamic state of a body for the next time step. Therefore, we want to introduce the most important numerical integration methods: semi-implicit Euler (also called symplectic Euler), Runge-Kutta methods and adaptive methods like the embedded Runge-Kutta.

In contrast to the well-known explicit Euler, the semi-implicit Euler uses the velocity at time t_0+h instead of time t_0 for the integration of the position vector:

$$\begin{aligned}\mathbf{u}(t_0+h) &= \mathbf{u}(t_0)+h\mathbf{M}^{-1}\mathbf{g}(\mathbf{q},\mathbf{u},t_0)\\ \mathbf{q}(t_0+h) &= \mathbf{q}(t_0)+h\mathbf{H}\mathbf{u}(t_0+h)\end{aligned}$$

where $\mathbf{M}$, $\mathbf{g}(\mathbf{q},\mathbf{u},t)$ and $\mathbf{H}$ are defined by equations 1, 14 and 15. The semi-implicit Euler is a first-order symplectic integrator. The advantage of integrating the velocities first is that the new velocities can be adapted before the position integration in order to resolve collisions or to simulate damping [GBF03, MHHR07].

Runge-Kutta methods are also very popular in the field of rigid body dynamics [BWAK03, RGL05, BS06, Ben07] for solving the initial value problem given by the equation of motion. An initial value problem is an ordinary differential equation

$$\dot{\mathbf{z}} = \mathbf{h}(t,\mathbf{z}(t))$$

together with an initial value $(t_0,\mathbf{z}_0)$. One of the most important methods in this field is the fourth-order Runge-Kutta:

$$\begin{aligned}\mathbf{k}_1 &= h\mathbf{h}(t_i,\ \mathbf{z}_i)\\ \mathbf{k}_2 &= h\mathbf{h}(t_i+\frac{1}{2}h,\ \mathbf{z}_i+\frac{1}{2}\mathbf{k}_1)\\ \mathbf{k}_3 &= h\mathbf{h}(t_i+\frac{1}{2}h,\ \mathbf{z}_i+\frac{1}{2}\mathbf{k}_2)\\ \mathbf{k}_4 &= h\mathbf{h}(t_i+h,\ \mathbf{z}_i+\mathbf{k}_3)\end{aligned}$$