

Realistic Animation of Rigid Bodies

James K. Hahn

Ohio Supercomputer Center and Department of Computer and Information Science
The Ohio State University
1224 Kinnear Road, Columbus, OH 43212

Abstract

The theoretical background and implementation for a computer animation system to model a general class of three dimensional dynamic processes for arbitrary rigid bodies is presented. The simulation of the dynamic interaction among rigid bodies takes into account various physical characteristics such as elasticity, friction, mass, and moment of inertia to produce rolling and sliding contacts. If a set of bodies is statically unstable, the system dynamically drives it toward a stable configuration while obeying the geometric constraints of the system including general *non-holonomic* constraints. The system also provides a physical environment with which objects animated using more traditional techniques can interact. The degree of interaction is easily controlled by the animator. A computationally efficient method to merge kinematics and dynamics for articulated rigid bodies to produce realistic motion is presented.

CR Categories and Subject Descriptors: I.3.7 [Three-Dimensional Graphics and Realism]: Animation

Additional Key Words and Phrases: Modeling, Simulation, Rigid bodies, Dynamics

1. Introduction

In traditional computer graphics, objects in a scene are looked upon as geometric shapes devoid of dynamical properties. The result is that the animator is forced to use his intuition about the physical world in planning the motion of objects in the scene. Since we are so sensitive to detecting anomalies in everyday physics and real motions tend to be complex, such techniques have generally proven unsatisfactory. Many animators have relied on specialized software

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

(usually ad hoc) to model specific types of motion. Analogous to the development of physically based illumination models in computer graphics display algorithms, we need to think of objects in a scene as real objects having mass, moment of inertia, elasticity, friction, etc.

We present a simulation system for computer animation capable of realistically modeling the dynamics of a general class of three dimensional motions of arbitrary rigid bodies. The system introduces the following concepts absent in previous dynamics simulation systems:

- Using collision analysis, the interaction among objects can be realistically simulated in a completely general way. This includes continuous contact and complex contact geometry, allowing the solution of general constraint problems including *non-holonomic* constraints. Collision analysis makes it possible to solve the constraint problems using only the absolute minimum quantities needed to describe motion, the first derivatives of position and orientation (velocity and angular velocity).
- The motion of articulated figures, whose limbs are under kinematic control of the animator, can be solved by simple conservation of momentum arguments. The algorithm is linear in the number of links moved and does not involve numerical solutions of differential equations involving joint forces or torques.
- The system provides a physically realistic environment with which objects animated using more traditional techniques, such as key-framing and forward and inverse kinematics, can interact.

2. Other Works in Dynamics Simulations

Recently there has been a rising interest in dynamics simulations in computer animation. Weil [18] has simulated the motion of cloth. Terzapoulos et al. [17] and Haumann [9] have investigated the dynamics of flexible objects in modeling their deformations.

In rigid body motion, the emphasis has been in simulating articulated figures. Armstrong and Green [1] and Wil-



helms and Barsky [19] have used dynamics to simulate human figures. Isaacs and Cohen [12] have used similar strategies but have also incorporated kinematic control of some of the joints. Two serious drawbacks of the dynamics approaches are the large computation time and the problem of control. Our method for articulated figure motion solves a slightly different problem of internal kinematic control of the limbs and external dynamic analysis. We feel that this is simpler and more effective in generating realistic motion in many cases.

The dynamics of contact between bodies that result in jointed mechanisms have been solved by researchers in studying articulated figures using simple constraints. This involves formulating a different solution for each configuration of constraints. Recently, Witkin et al. [20] have investigated purely geometric constraints to deal with several classes of constraint problems. Barzel and Barr [2] have used a similar strategy but in the context of dynamics simulations. This works well for animating the process of satisfying the constraints and is more general in the sense that different configurations of constraint do not need different solutions. However it is still specialized in the sense that for each different class of constraint one must be able to define the constraint in the form of an equation relating the coordinates and formulate a different solution. Other systems have modeled contacts between objects directly using repulsive forces [17, 20] and by using imaginary springs and dampers [19]. Our model takes a more rigorous and completely general approach that is able to realistically simulate friction (especially the transition between sliding and sticking contact) and elastic properties (especially the transition between impact and continuous contact) of any contact between arbitrary rigid bodies.

3. Overview of the System

The system was implemented on the Symbolics 3600 family of machines using Common LISP with Flavors. The general flowchart is given in Figure 1.

Each object in the scene to be simulated is given physical characteristics such as shape, density, coefficient of restitution, coefficient of friction, and link hierarchy if any. From these, other properties such as total mass, center of mass, moment of inertia tensor, and *principal axes* are calculated.

The dynamic state of each object include the linear and angular velocities, the position, and the orientation. The current dynamic state is used to solve for the dynamic state at an infinitesimal time increment dt later. Usually no oversampling (more than 30 per second video and 24 per second film) is necessary and dt represents the time increment between frames of the animation. The update is done in two steps. First, the objects are moved using the current dynamic state. This involves solving for the positions and orientations of the objects using self-starting numerical solutions of sets of coupled first order differential equations (such as

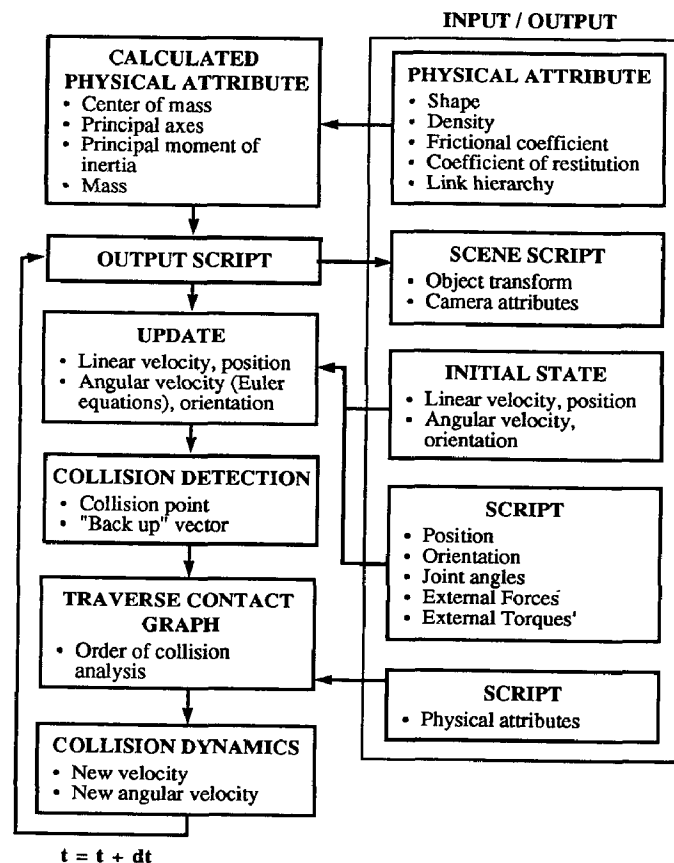


Figure 1. Flowchart of the system

Runge-Kutta [10]). If the motion of the object is "scripted" then the state is read in. Second, the objects are checked for intersections. If contact occurred, the new dynamic state for the objects that were effected are calculated using impact dynamics in an order given by traversing the "contact graph".

4. Motion of Rigid Bodies Under External Forces and Torques

The general motion of a rigid body can be decomposed into a linear motion of a point mass equal to that of the body located at the center of mass of the body under an external force and a rotational motion about the center of mass under an external torque. The linear motion under a force F can be calculated by solving the set of coupled differential equations

$$\begin{aligned} \dot{\mathbf{V}} &= \mathbf{F} / m \\ \dot{\mathbf{X}} &= \mathbf{V} \end{aligned} \quad (1)$$

where m is the mass of the object, \mathbf{X} is the position vector, and \mathbf{V} is the linear velocity.

4.1 Euler Equations

If we choose to solve the rotational dynamics in a body fixed coordinate system given by a set of axes known as the

principal axes, a set of simplified coupled first order differential equations for the angular velocity \mathbf{W} , known as the *Euler equations*, results (Appendix B). These along with

$$\begin{aligned}\dot{\mathbf{T}}_x &= \mathbf{W}_x \\ \dot{\mathbf{T}}_y &= \mathbf{W}_y \\ \dot{\mathbf{T}}_z &= \mathbf{W}_z\end{aligned}\quad (2)$$

where \mathbf{T} 's are the orientations in the principal axes coordinate, form a set which can be solved by a numerical technique such as Runge-Kutta [10]. Analytic solutions also exist for the equations in which Jacobian elliptic trigonometric functions are used [11]. These solutions may have an advantage when efficient packages are available for elliptic functions [14]. The analytic solutions could also make motion planning easier.

4.2 Moment of Inertia Tensor and the Principal Axes

The moment of inertia tensor for a rigid body, which is the rotational analogue of mass, is given by [16]

$$\mathbf{I} = \int \rho (\mathbf{R}^2 \mathbf{U} - \mathbf{R} \mathbf{R}) dv \quad (3)$$

where ρ is the density of the object, \mathbf{R} is the location vector of the volume element dv , \mathbf{U} is the unity dyadic, and $\mathbf{R} \mathbf{R}$ is a dyad product. This is a *Hermitean* or a symmetric tensor of the second rank. The 3×3 matrix form of \mathbf{I} is given in Appendix A. The inertia tensor can be transformed under rotation and translation to any coordinate system (Appendix A). The numerical integrations for the inertia tensor are performed in a coordinate frame where the origin coincides with the center of mass of the object to facilitate the separation of linear and rotational motion.

We can rotationally transform the inertia tensor to a coordinate frame in which the tensor is diagonalized. The existence of such a coordinate frame for any inertia tensor is guaranteed by the fact that it is a Hermitean [7]. This is just the problem of finding the eigenvectors \mathbf{E} and its associated eigenvalues i for the matrix \mathbf{I} .

$$\mathbf{I} \cdot \mathbf{E}_j = i_j \cdot \mathbf{E}_j \quad (1 \leq j \leq 3) \quad (4)$$

A numerical solution such as the power method [10] can be used to find the largest and the smallest eigenvalues and the associated eigenvectors. The \mathbf{E} 's which are the axes of the coordinate system in which \mathbf{I} is diagonalized are known as the *principal axes* and the i 's which are the diagonal elements of the tensor in the principal axes coordinate are known as the *principal moments of inertia*. Intuitively the principal axes correspond to the "axes of symmetry" of an object and the principal moments of inertia corresponds to the associated moments of inertia.

It is important to realize that for any arbitrary shaped

object, one can find the principal axes and the principal moments of inertia. They are invariant geometrical descriptions of the object in the body fixed coordinate system and need to be calculated only once for each object.

4.3 Dynamics of Articulated Figures

In most interesting human or animal motion, all the joints are under some autonomous control. Cases where limbs react to other limb motion or external forces and torques with absolutely no internal muscle control are rare. Also, empirical studies have shown that the unrestrained human limb motion are determined by intelligent trajectory planning in purely kinematic terms [4]. Even constrained systems in which all the parts cannot be defined as a mechanical system (e.g. where joints are controlled by feedback systems consisting of muscles and sensors) cannot be modeled using pure dynamics. Therefore the computationally intensive solution for the internal dynamics of articulated figures may be unjustified except for modeling inanimate jointed objects. We use kinematics to control joint trajectories [5] and dynamics to model the effects of limb motion and external forces and torques on the body as a whole.

The articulated body is defined in the form of an arbitrary tree of links. The tree structure makes the kinematic control easier but is not necessary for the dynamics analysis. In fact any system in which quanta of masses are moved within the body by the animator, such as an object changing shape, can use approaches similar to the following.

The dynamics of the figure as a whole proceeds as in the treatment of a single rigid body. The total moment of inertia tensor is calculated by summing the inertia tensors of individual links in the world coordinate and then transforming to the body fixed coordinate system with its origin at the center of mass of the articulated body. The principal axes are calculated from the total inertia tensor.

When the joints are being driven kinematically, there are additional motions of the whole object in the body fixed reference frame due to the conservation of linear and angular momentum of the system. When a joint j is moved with angular velocity $d\mathbf{W}_j$ in the joint fixed coordinate system, (Figure 2) the whole body must rotate by

$$d\mathbf{W}' = -\mathbf{I}_1^T [(\sum_i \mathbf{I}_i) d\mathbf{W}_j + (\mathbf{R}_j \times (d\mathbf{W}_j \times \mathbf{C}_j)) (\sum_i m_i)] \quad (5)$$

in the body fixed coordinate system to conserve angular momentum. \mathbf{I}_1^T is the transpose of the total inertia tensor of the whole body, the first sum is the moment of inertia tensor associated with the joint (i.e. the inertia tensors of all the descendants of j) in the joint coordinate system, \mathbf{R}_j is the location of the joint with respect to the body fixed coordinate system, \mathbf{C}_j is the location of the center of mass of the descendants of the joint in the body fixed coordinate system, and the last sum is the total mass of the descendants.

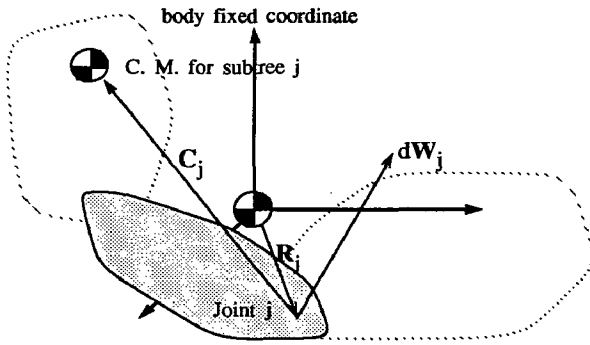


Figure 2. Moving joint j by dW_j in the body fixed coordinate

The linear momentum is conserved by finding the new center of mass of the whole body after the change and moving the body so that the center of mass coincides with the origin of the body fixed frame.

The above analysis is done at each time step for the joints that are being driven. The order in which the joints are treated is given by traversing the tree from the leaves toward the root. Then the inertia tensor of a subtree can be used to calculate the inertia tensor of its ancestor joint.

There is a new angular velocity of the entire body due to the change in the inertia tensor

$$W_t' = \tilde{I}_t' \cdot L_t \tag{6}$$

where \tilde{I}_t' is the matrix transpose of the new total inertia tensor and L_t is the invariant total angular momentum of the body.

5. Interaction Among Objects

Dynamics simulations (and disciplines involved in solving mechanical problems in general) have used the concept of *holonomic constraints* to solve the dynamics of continuous contact between objects. Holonomic constraints are a class of constraints in which one can define the constraints in the form of an equation of coordinates [7, 16]. They are abstractions of a small subset of the general constraint introduced to facilitate an analytic solution. The majority of constraints do not have such simple abstractions. As an example, the constraints of the links in a chain cannot be abstracted as, for example, a 3-degrees of freedom ball and socket joint. In fact each link has 6 degrees of freedom until they are in contact with other links and then the interaction is very complex (Figure 13). The most general constraint should therefore not be expressed in terms of what the objects *must* do (remain on a point, a line, etc.) but in terms of what the objects *must not* do (penetrate each other).

In the system, interactions among objects are simulated using collision detection to model the general constraints and impact analysis to solve the dynamics. Impact analysis

has been extended to include continuous contacts and simultaneous contact of many bodies. This makes it possible to solve the dynamics of arbitrary interactions without solving differential equations involving finite forces and torques. It also makes possible the realistic simulation of the transition between instantaneous and continuous contacts and between sticking and sliding contacts.

5.1 Impact Dynamics

An analysis of the impact process of two rigid bodies was proposed by Routh in the late nineteenth century [15]. Routh included the effect of the *Coulomb* model of friction and partially elastic materials employing graphical solution methods. His work remains essentially unchanged in modern expositions [6]. The following uses analytic and numerical solution techniques involving only the states before and after the collision.

When two bodies collide and the contact area of one of the bodies is locally planar, one can define the normal N to the tangent surface of contact between the two bodies (Figure 3). If a surface of contact cannot be defined for the impact (for example when a point of one object strikes the point of another) the outcome is theoretically indeterminate. In the system, we average the normals of neighboring polygons. In the following, "normal" refers to N and "tangential" refers to the direction along the tangent surface of contact.

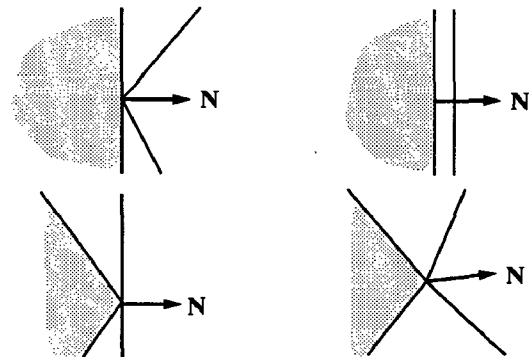


Figure 3. Modes and normals of local contact

In an impact process, the bodies act on each other with *impulse P*

$$P = \int_{\Delta t} F dt \tag{7}$$

where F is the large force between the objects that act through an infinitesimally short time interval Δt . Since the integral is over time as Δt approaches 0, any finite forces such as gravity does not contribute to P . The conservation of linear and angular momenta gives (Figure 4)

$$\begin{aligned} m_1 (V_1' - V_1) &= -P_1 \\ m_2 (V_2' - V_2) &= +P_2 \end{aligned} \tag{8a}$$

$$\begin{aligned} I_1 \cdot (W'_1 - W_1) &= R_1 \times (-P_1) \\ I_2 \cdot (W'_2 - W_2) &= R_2 \times (+P_2) \end{aligned} \quad (8b)$$

The subscripts stand for each of the bodies and the primes denote the quantities after the impact. The moment of inertia tensors and the angular velocities are transformed from the body fixed frames to the world coordinate frame before they are used in these equations.

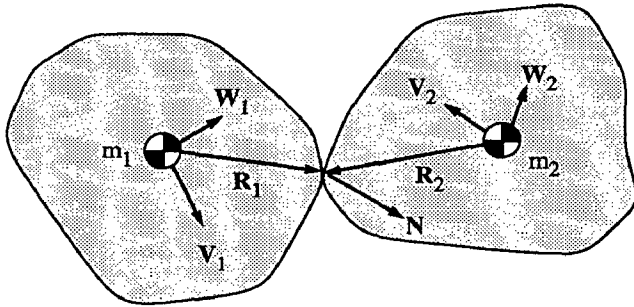


Figure 4. Collision between two objects

Here we introduce an empirical result known as the *generalized Newton's rule* [6].

$$\frac{(V'_1 + W'_1 \times R_1) \cdot N - (V'_2 + W'_2 \times R_2) \cdot N}{(V_1 + W_1 \times R_1) \cdot N - (V_2 + W_2 \times R_2) \cdot N} = -\epsilon \quad (9)$$

The constant of proportionality, ϵ , known as the *coefficient of restitution* depends to a large extent on the elasticity of the materials of the two constituent objects. ϵ has a value ranging from 0, corresponding to a perfectly inelastic collision to a value of 1, corresponding to a perfectly elastic collision where no kinetic energy is lost.

Now we consider friction between the two bodies at the moment of impact. Coulomb's law states [13]

$$|F_t| \leq \mu F_n \quad (10)$$

where t is the tangential component and n is the normal component of the force F between the objects. The positive number μ is the coefficient of friction and depends solely on the materials of the two bodies. When $\mu = 0$ the interaction is frictionless. If the two objects are moving tangentially relative to each other at the point of contact then the equality holds in (10). We first assume that the two bodies do not slip on impact at the point of impact. Then

$$\begin{aligned} [(V'_1 + W'_1 \times R_1) - (V'_2 + W'_2 \times R_2)]_t &= 0 \\ [(V'_1 + W'_1 \times R_1) - (V'_2 + W'_2 \times R_2)]_r &= 0 \end{aligned} \quad (11)$$

where t and r are the orthogonal components of the velocity vector perpendicular to N . Equations (8), (9), and (11) give us 15 independent equations in 15 unknowns (P and for both

objects V' and W'). If the solution for P satisfies

$$|N \times (P \times N)| \geq \mu |P \cdot N| \quad (12)$$

then by (10) the no slip assumption that leads to (11) is not valid and the two bodies are sliding at the point of contact. In this case

$$\begin{aligned} P_t &= 0 \\ P_r &= \mu |P \cdot N| \end{aligned} \quad (13)$$

where t is the direction given by $(P \times N)$ and r is the direction given by $N \times (P \times N)$, can be substituted for (11). The new set of 15 equations can be solved for the unknowns.

V' and W' for both objects constitute the new state after the collision. For the collision of a rigid body with an object of infinite mass (e. g. floor or other objects that effect the environment but are not in turn effected by it) the development is similar except only the conservation of momentum and energy of the one body is considered resulting in 9 equations in 9 unknowns (P , V' , and W' for the body).

5.2 Continuous Contact

After a collision analysis is performed between two objects, if the relative velocity of the objects at the collision point in the direction of the normal of local contact is less than a small threshold then the objects can be considered to be in continuous contact. If the objects are applying a force on each other, for example when one of the objects is the floor, the dynamics can still be simulated using the impact equations (8), (9), (11) and (13). Although the forces and torques do not appear explicitly, their contribution is seen as the gain in momenta of the object during dt which are the impulses that the support applies to the object in the impact process. Even though we are approximating continuous contact with a series of instantaneous contacts, oversampling is usually not necessary.

5.3 Collision Detection

In order to minimize the number of polygon to polygon intersection tests, a hierarchical method involving bounding boxes is used. At the bottom of the hierarchy, each edge of object 2 is tested against each polygon of object 1 and vice versa. Since the collision detection is performed at discrete intervals of time, two types of penetrations are possible [3].

In Figure 5, the collision point is given by the "inside" vertices of the edges that intersect polygons. Assuming that object 1 is stationary in world coordinate one can define a ray originating from the collision point in a direction given by the relative velocity of the two objects at the collision point. If we assume that the velocity of the two objects at the collision point remains constant during the time step, the ray represents the path that the collision point of object 2 took when it penetrated object 1. The intersection between the ray and the polygons of object 1 represents the actual pene-

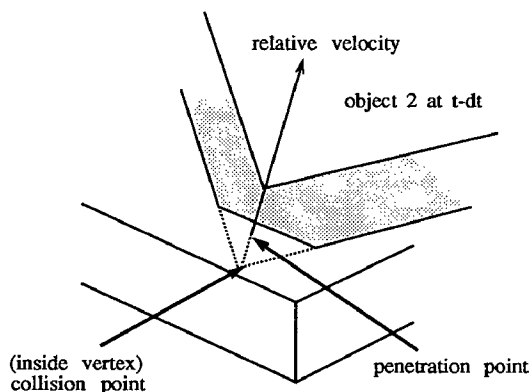


Figure 5. Point collision detection

tration point at which the two objects collided. In order to determine the actual positions and orientations of the objects at time t , both objects must be backed up to the time of the collision and collision dynamics used to generate the new velocities and thus the positions and the orientations at t . In our implementation, assuming sufficiently small dt , object 2 is "backed up" at time t so that the two objects touch but do not penetrate.

Figure 6 represents an edge collision where an edge penetrates more than one polygon. The penetration point is calculated by intersecting the polygon swept by the edge during the time step with the edges of the pierced polygons. The collision point is calculated by finding the intersection between the penetrating edge and a ray originating from the penetration point in a direction given by the negative of the relative velocity.

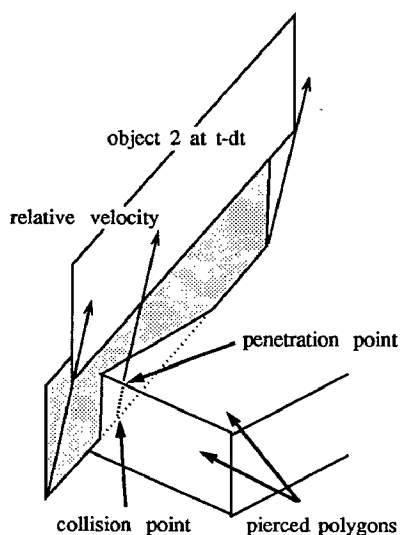


Figure 6. Edge collision detection

The above algorithm assumes that the time step dt and/or the velocity of the polygons of the objects are small enough such that the distance covered during dt is much smaller than the dimensions of the polygons. If this were not

the case then one must consider the volume swept by the bounding boxes and the polygons during dt in the collision detection algorithm not to miss the collision.

5.4 Complex Contact Geometry

For a scene consisting of a number of bodies in simultaneous contact, the impact analysis is applied to each pair of objects. If an object belongs to more than one such pair, then the contributions from each of the impacts are summed. An arbitrary order in handling the contact pairs could result in penetration between objects even after the collision analysis because of the necessity to "back up" after an impact is discovered.

In general, the contact geometry of a scene can be represented as a graph (Figure 7). The nodes represent the

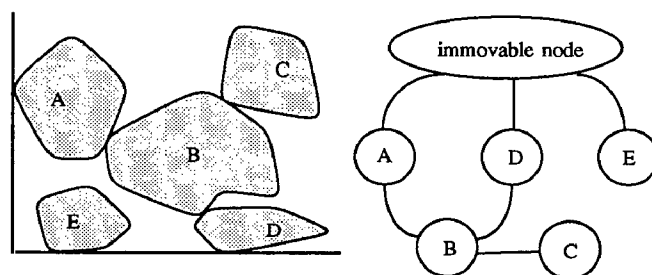


Figure 7. Contact geometry and associated graph

objects and the edges represent contacts. The special immovable node represents the set of all immovable objects in the scene. In order to prevent objects from being "backed up" into immovable objects, the contact pairs are handled in an order given by a breadth first search of the graph starting from the immovable node. If there are no immovable nodes, then the breadth first search can start from any node. In graphs involving cycles the last contact pair to be considered in the cycle must be checked so that the objects are not "backed up" into each other.

6. Control Issues

In any simulation system, there is a trade-off between automation and control by the animator. In our system the animator has a great deal of flexibility in determining how much of a control he has on the motions of objects in the scene. He can take full advantage of automation by specifying the initial state of the system and letting the system generate the subsequent motions. Through a series of experimentation, the desired overall motion can be achieved in a relatively short period of time. Our experiences have shown that most sequences (averaging approximately 10 seconds of animation) can be generated after only a few trials. By controlling the physical characteristics of the object (possibly as a function of time) the animator can manipulate the physics to his liking.

Some of the objects can be under the direct control of the animator allowing the integration of traditional animation techniques with simulation. The trajectories of these objects are usually scripted by specifying the position and orientation of the objects as a function of the frame number. In order to apply impact dynamics, the script is numerically differentiated to get the velocities as a function of time. A weighted average of the scripted velocities and the velocities given by the impact dynamics (if any) can be used to move the object. The amount of effect that the other objects have on the object being moved is controlled by scaling the mass and the moment of inertia of the object being moved. With a high scaling factor, the object effects the environment more and follow closely the scripted trajectory.

Some of the objects can be moved by applying forces and torques directly. Control is harder since the effect on an object is not clear. Inverse dynamics [12] can be used to calculate in advance the forces and torques needed to move an object in a desired trajectory .

7. Examples of Animation

Figures 9-16 illustrates a few of the wide range of realistic animation sequences generated with the system [8]. The collision detection algorithm, which is of order n^2 in the number of polygons when the objects are close enough, takes most of the calculation time in most sequences. Figure 8 gives approximate calculation times per frame with and without collision detection.

Animation	Without collision detection	With collision detection
Figure 9	.05	1.33
Figure 10	.05	6.97
Figure 11	.50	0.54
Figure 12	.60	3.41
Figure 13	.15	6.31
Figure 14	.07	8.60
Figure 15	.15	0.59
Figure 16	.04	1.52

Figure 8. Calculation time per frame in seconds

Figure 9 illustrates how the animator can direct the overall motion (even fairly complex motion). We wanted the car to come off the ramp, bounce off of a second car, and then crash into a third car. The low level details of the motion were generated by the system. Figure 10 (showing procession and nutation of the tops) illustrates how the laws of physics can be manipulated to get a desired effect. We wanted the tops to spin slowly to prevent temporal aliasing. In order to keep the tops from falling over because of a small angular velocity, the moments of inertia were increased artificially. In the articulated figure animation (Figure 11), the joint trajectories were kinematically specified and the system generated the external motion. In Figure 12, the motion of the rocket was scripted by the animator. The plate pieces realistically react to being pushed by the rocket. In the colli-

sion analysis, the mass and the moment of inertia of the rocket were set essentially to infinity compared to those of the plate pieces. In Figure 13, the top link is fixed. The other links in the chain are free to move without artificially imposed constraints. The chain was released from an initial state. The motion reflects the fact that the links are faceted and have some friction. Figures 14, 15, and 16 give examples of continuous and complex configuration of contacts among arbitrary shaped objects.

8. Conclusion

We are currently studying the possibility of using various strategies to search the space spanned by the dynamic state of an object to maximize or minimize certain quantities such as the closeness to a desired trajectory for an object. This type of motion planning would make it possible to precisely control the animation using our system. We are also in the process of implementing a near linear order collision detection algorithms employing neighborhood or spatial information.

In the system, the dynamics simulation proceeds from an initial state by a time series analysis of the linear and rotational motion. The rotational dynamics is simplified by solving the Euler equations in the principal axis reference frame. The external dynamics of kinematically driven articulated figures are solved principally by using conservation of momenta. The complex interactions between objects (including continuous contacts) are modeled in a completely general and novel way using collision detection and impact dynamics. The use of conservation of momenta in the dynamics analysis allows the solution for the motion involving only the velocities and not the accelerations. Scripts can be specified to integrate simulation and traditional animation techniques and to influence the dynamics.

The variety of realistic animations generated with the system show promise in systems based on physical simulations becoming an integral part of animation of rigid bodies.

Acknowledgments

I would like to thank Rick Parent, Brian Guenter, Michael Girard, and everyone in the "Symbolics Group" at ACCAD for many helpful discussions and software support. I would also like to thank Chuck Csuri and Tom Linehan for providing a free and supportive research atmosphere without which this work would not have been possible. This research was supported in part by a National Science Foundation grant DCR-8304185.

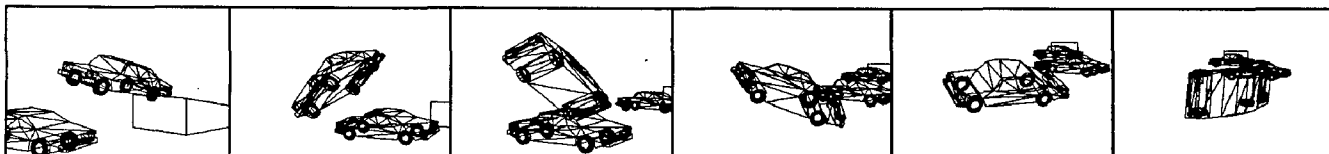


Figure 9.

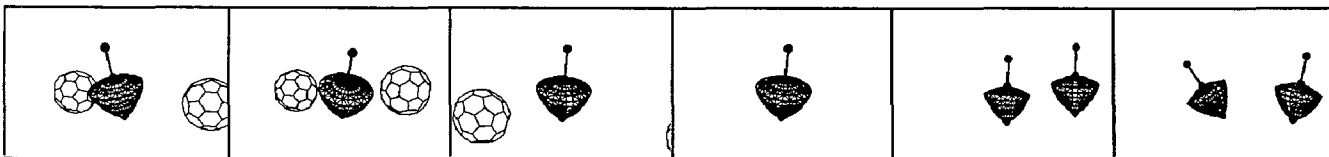


Figure 10.

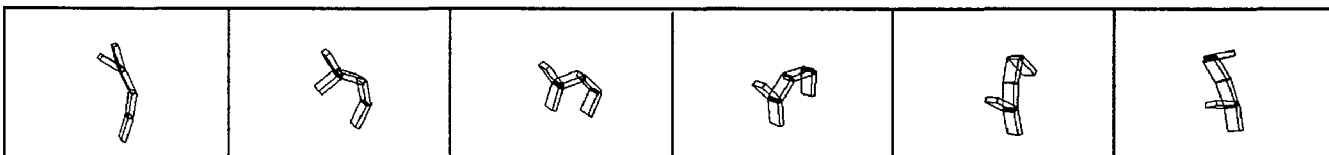


Figure 11.

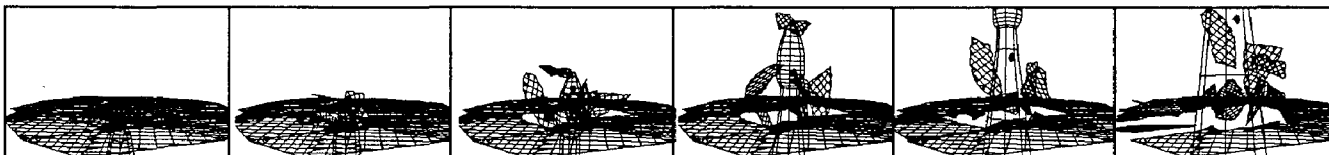


Figure. 12

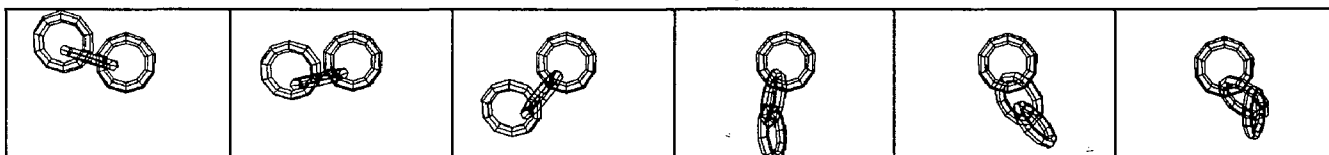


Figure. 13

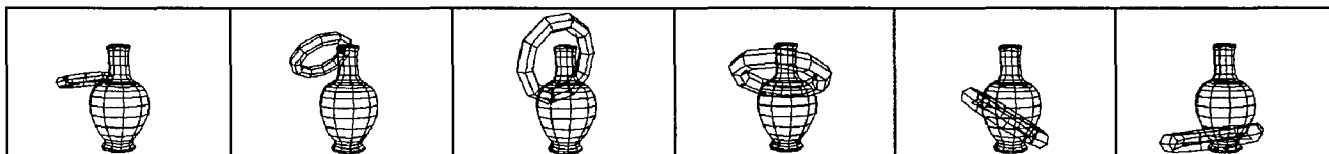


Figure. 14

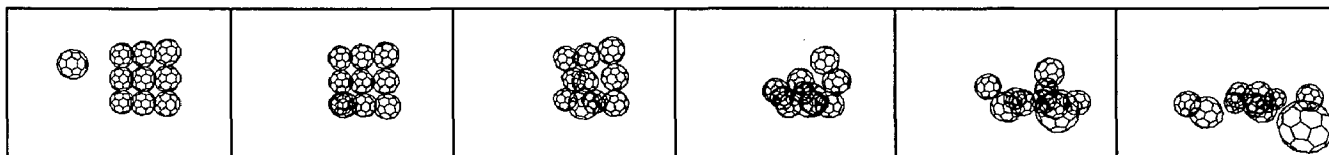


Figure. 15

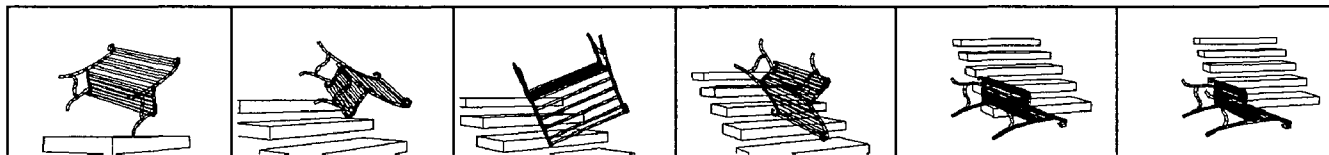


Figure. 16

Appendix A

In the 3 x 3 matrix form of the moment of inertia tensor of an object, the diagonal elements or the *moment of inertia coefficients* are given by [7, 16]

$$\begin{aligned} I_{xx} &= \int \rho (y^2 + z^2) dv \\ I_{yy} &= \int \rho (x^2 + z^2) dv \\ I_{zz} &= \int \rho (x^2 + y^2) dv \end{aligned} \quad (A1)$$

and the off-diagonal elements or the *product of inertia* are given by

$$\begin{aligned} I_{xy} &= \int \rho (x y) dv \\ I_{xz} &= \int \rho (x z) dv \\ I_{yz} &= \int \rho (y z) dv \end{aligned} \quad (A2)$$

where the subscripts stand for the matrix indices, ρ is the density of the object, and the integral is over the volume of the object. The inertia tensor can be transformed to any coordinate frame. The matrix of the tensor transforms under a 3 x 3 orthogonal matrix A as a *similarity transform* [7]

$$I' = A I A^T \quad (A3)$$

where the superscript T stands for the matrix transpose. For our purposes of rotational transform the matrix A is the direction cosine matrix of the two coordinates. For the translational transform, by the matrix form of the tensor (A1) and (A2)

$$\begin{aligned} I'_{xx} &= I_{xx} + m (y^2 + z^2) \\ I'_{yy} &= I_{yy} + m (x^2 + z^2) \\ I'_{zz} &= I_{zz} + m (x^2 + y^2) \\ I'_{xy} &= I_{xy} + m (x y) \\ I'_{xz} &= I_{xz} + m (x z) \\ I'_{yz} &= I_{yz} + m (y z) \end{aligned} \quad (A4)$$

where x , y , and z gives the translation of the coordinate frame and m is the mass of the object.

Appendix B

The rotational dynamics is given by

$$dL/dt = N \quad (B1)$$

$$L = I \cdot W \quad (B2)$$

where L is the angular momentum of the body, N is the external torque being applied to the body, I is the moment of inertia tensor of the body, and W is the angular velocity of the body that we want to solve for. The major obstacle for a simple solution is that the rotational equivalent of mass, the

moment of inertia tensor, is not constant with respect to an inertial reference frame but changes as the body rotates. This can be avoided by solving the rotational dynamics in a frame that is fixed in the body. Taking the time derivative of (B1) with reference to a coordinate frame fixed in the body

$$dL/dt + W \times L = N \quad (B3)$$

Substituting (B2) into (B3)

$$I \cdot dW/dt + W \times (I \cdot W) = N \quad (B4)$$

If we choose the principal axes as the body fixed axes and transform (B4) to the body fixed coordinate we get a set of simplified differential equations

$$\begin{aligned} I_x \dot{W}_x + (I_z - I_y) W_z W_y &= N_x \\ I_y \dot{W}_y + (I_x - I_z) W_x W_z &= N_y \\ I_z \dot{W}_z + (I_y - I_x) W_y W_x &= N_z \end{aligned} \quad (B5)$$

where the products of inertia do not appear. These are known as the *Euler equations* [7]. The x , y , and z are the directions of the principal axes.

References

1. Armstrong, William W. and Green, Mark W.. "The Dynamics of Articulated Rigid Bodies for Purposes of Animation". *Proc. Graphics Interface 85* (1985), pp. 407-415.
2. Barzel, Ronen and Barr, Alan H.. "Modeling With Dynamic Constraints". *ACM SIGGRAPH '87 Course Notes* (1987).
3. Boyse, John W.. "Interference Detection Among Solids and Surfaces". *Communications of the ACM* 22, (January 1979), pp. 3-9.
4. Flash, Tamar and Hogan, Neville. "The Coordination of Arm Movements An Experimentally Confirmed Mathematical Model". *MIT A.I. Memo 786* (November 1984).
5. Girard, Michael and Maciejewski, A. A.. "Computational Modeling for the Computer Animation of Legged Figures". *Computer Graphics* 19, 3 (July 1985), pp. 263-270.
6. Goldsmith, Werner. *Impact: The Theory and Physical Behavior of Colliding Solids*. Edward Arnold Pub. Ltd., London, 1960.
7. Goldstein, Herbert. *Classical Mechanics*. Addison-Wesley Publishing Company, Reading, MA, 1950.
8. Hahn, James. "Rigid Body Dynamics Simulations". *ACM SIGGRAPH '87 Film and Video Show* (1987).
9. Haumann, David. "Modeling the Physical Behavior of



- Flexible Objects". *ACM SIGGRAPH' 87 Course Notes* (1987).
10. Hornbeck, Robert W.. *Numerical Methods*. Quantum Publishers, Inc., New York, NY, 1975.
 11. Hughes, Peter C.. *Spacecraft Attitude Dynamics*. John Wiley & Sons, Inc., 1986.
 12. Isaacs, Paul M. and Cohen, Michael F.. "Controlling Dynamic Simulation With Kinematic Constraints, Behavior Functions and Inverse Dynamics". *Computer Graphics 21, 4* (July 1987), pp. 215-224.
 13. Kane, Thomas and Levinson, David. *Dynamics: Theory and Applications*. McGraw-Hill, New York, N.Y., 1985.
 14. Press, William H.. *Numerical Recipes*. Cambridge University Press, 1986.
 15. Routh, E. J.. *Dynamics of a System of Rigid Bodies*. Macmillan and Company, Ltd., London, 1905.
 16. Symon, Keith R.. *Mechanics*. Addison-Wesley Publishing Company, Reading, MA, 1971.
 17. Terzopoulos, Demetri, Platt, John, Barr, Alan, and Fleischer, Kurt. "Elastically Deformable Models". *Computer Graphics 21, 4* (July 1987), pp. 205-214.
 18. Weil, J.. "The Synthesis of Cloth Objects". *Computer Graphics 20, 4* (August 1986), pp. 49-54.
 19. Wilhelms, J. and Barsky, B.A.. "Using Dynamic Analysis for the Animation of Articulated Bodies Such as Humans and Robots". *Proc. Graphics Interface 85* (May 1985), pp. 97-104.
 20. Witkin, Andrew, Fleischer, Kurt, and Barr, Alan. "Energy Constraints On Parameterized Models". *Computer Graphics 21, 4* (July 1987), pp. 225-229.