Proceedings of the 2000 IEEE/RSJ
International Conference on
Intelligent Robots and Systems

A Framework for Multi-Contact Multi-Body Dynamic Simulation and Haptic Display

Diego Ruspini and Oussama Khatib
Robotics Laboratory, Department of Computer Science
Stanford University, Stanford CA, 94305
ruspini@cs.stanford.edu, khatib@cs.stanford.edu

Abstract

We present a general framework for the dynamic simulation and haptic exploration of complex virtual environments. This work builds on previous developments in simulation, haptics, and operational space control. The relations between the dynamic models used in simulation and the models originally developed for robotic control are also presented. This framework has been used to develop a simulator that can model complex interaction between generalized articulated mechanical systems and permit direct "hands-on" interaction with the virtual environment through a haptic interface.

1 Introduction

In recent years, there have been many efforts to accurately simulate physical environments in both robotics and computer graphics. A physically accurate simulation can be used to obtain insight into the real-world behavior of a robotic, manufacturing, space or other dynamic environment. Haptics is one area where the need to find the dynamic motions of a virtual environment rapidly is particularly important.

In haptics, a force reflecting mechanical device is used to apply forces to a user, (typically through the user's finger or hand) and thus, create the illusion of physical contact with a real physical environment. To deliver a convincing and intuitive sense of presence, the motions of the virtual model must behave realistically as they are influenced by the forces applied by the user. In this way the user can easily obtain information about an object's size, shape, effective mass, stiffness, as well as many other internal and external object properties. In this paper we discuss our effort to develop a general purpose dynamic virtual environment allowing direct "hands-on" interaction through a haptic interface.

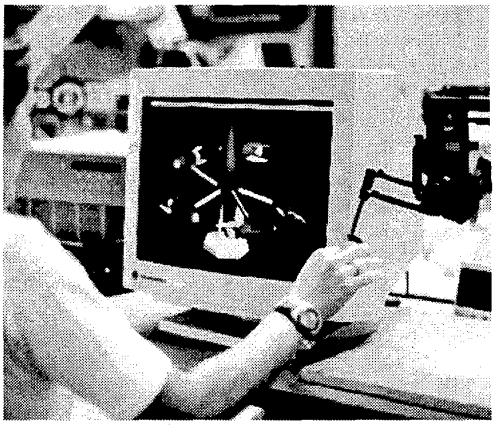

Figure 1: A user haptically interacting with a dynamic virtual environment (left).

2 Haptic Rendering

Early haptic rendering systems modeled surface contacts by generating a repulsive force proportional to the amount of penetration into an obstacle. While these penalty based methods, worked well to model simple obstacles, such as planes or spheres, a number of difficulties were encountered when trying to extend these models to display more complex environments.

An alternative is not to look at the penetration of the user's finger into the object at all, but instead to constrain the motions of a substitute virtual object. In the method we proposed [12], a representative object, a "proxy," substitutes in the virtual environment for the physical finger or probe. The "virtual proxy" can be viewed as if connected to the user's real finger by a stiff spring. As the user moves his/her finger in the workspace of the haptic device he/she may pass into or through one or more of the virtual obstacles. The proxy, however, is stopped by the obstacles

and quickly moves to a position that minimizes its distance to the user's finger position. The haptic device is used to generate the forces of the virtual spring which appears to the user as the constraint forces caused by contact with a real environment. This approach is similar to the method for the "gob-object" first proposed by Zilles et. al [15] but does not require apriori knowledge of the surface topology. An example of the system can be seen in Fig. 1.

3 Dynamic Motion Models

In the system described previously [12], only interactions between a simple representative object were considered. Inter-object interactions were not modeled. Considerable work in modeling interactions between multiple simple rigid bodies has been conducted [2] [1] [8]. Most of these systems, however, are too slow for interactive simulation and can not model articulated linkages efficiently.

Efficiency can be improved by modeling only the true degrees of freedom of the system and avoid solving for the internal constraints of the system. The configuration of an n-d.o.f. object can be described by q, a set of n independent generalized coordinates. The equation of motion for the system can be described by

$$\ddot{q} = A^{-1}(q)[\Gamma - b(q,\dot{q}) - g(q)] \tag{1}$$

where $A(q)^{-1}$ designates the inverse kinetic energy matrix, $b(q,\dot{q})$ the centrifugal and Coriolis force vector, $g(q)$ the gravity force vector, and Γ the generalized torque vector of the object. Several methods of computing the inverse equation of motion for a set of rigid bodies have been proposed [3].

When two or more objects exist in an environment their configuration vectors can be combined into a new configuration vector $q = [q_1^T q_2^T \ldots q_l^T]^T$ where l is the number of independent objects in the environment and q_i is the configuration vector for the ith object. Because the equations for a set of objects have the same form as equations for a single object, the entire system can be treated as a single n degree of freedom body where n is the sum of the degrees of freedom of each object.

When contact or collision occurs, between objects or links in the system, a constraint exists and one or more of the terms in q is no longer independent. In this case a set of constraint forces or impulses must be applied to prevent inter-penetrations between the primitives.

4 Contact Space

Given two bodies A, B in collision there exists a set of points c_a on body A and c_b on body B such that at collision time t, $c_a(t) = c_b(t)$. If the objects in the world are modeled as being constructed from the union of convex polyhedron, the contact region between two bodies will be defined by a set of convex polygons. In this situation it is sufficient to consider contact only at the extremal-points of the contact region created by the intersection of the contact surfaces [9].

Given this polygonal contact region assumption, only a finite number of contact points need to be considered. We shall define c_{a_i} as the i^{th} contact point attached to body A. In addition for each contact point a unit normal n_i perpendicular to the contact region is defined. This information can be used to define a set of contact parameters x_i that describe the relative distance between contact point c_{a_i} and c_{b_i} in the direction perpendicular to the contact surface:

$$x_i = n_i^T(c_{a_i} - c_{b_i}), \tag{2}$$

for $1 \leq i \leq m$. These "contact" parameters form a space that describes the motion of the bodies during contact and collision. An example of this space is illustrated in Fig. 2. For clarity, this space is shown when the objects are separating, in reality the space is only defined in the immediate neighborhood around the time of contact.

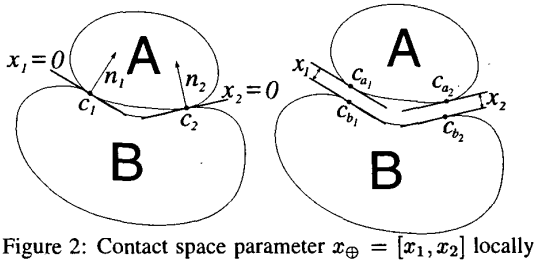

Figure 2: Contact space parameter $x_\oplus = [x_1, x_2]$ locally describe the motion at the contact normal to the surface.

The contact space parameters will in general not be independent. Consider when a four legged rigid table rests on a flat surface. The position of any one of the legs can be inferred by knowing the positions of the other three. Additionally, because of the rigidity condition at most three forces at the contact points are required to prevent the table from penetrating the floor. The force on one of the legs will be zero.

We will therefore define two sets of contact space parameters x and $x_\oplus$. The vector $x_\oplus$ consists of the full "augmented" set of contact space parameters (one per contact point). A subset of the contact parameters $x \subseteq x_\oplus$ contains only the "active" contact points of the entire contact

space. A contact point is "active" if the force or impulse applied at the contact point is non-zero. Which parameters belong to the "active" subset is for now unknown, but will be identified later.

We will associate a matrix S such that $x = \mathrm{S}x_\oplus$ that selects the members of $x\oplus$ that belong to x. Given these spaces we can define other parameters for velocity $v_\oplus = \dot{x}_\oplus$, acceleration $a_\oplus = \ddot{x}_\oplus$, forces $f_\oplus = \mathrm{S}f$, and impulses $p_\oplus = \mathrm{S}p$.

At the time of contact/collision t, $x_\oplus = \vec{0}$ and a Jacobian $v_\oplus = J_\oplus \dot{q}$ can be found [10]. In addition it possible to define the augmented operational (contact) inertia matrix:

$$\Lambda_\oplus^{-1} = J_\oplus A^{-1} {J_\oplus}^T. \tag{3}$$

This matrix is very similar to the operational space inertia matrix used in robotic control [7]. The matrix $\Lambda_\oplus$ represents the effective mass seen at all the contact points and characterizes the dynamic relationships between the contact points. The inverse active contact space inertia matrix

$$\Lambda^{-1} = \mathrm{S}\Lambda_\oplus^{-1}\mathrm{S}^T \tag{4}$$

being composed from a set of independent contact parameters is positive definite and hence invertible.

5 Impulse Force Resolution

If two or more bodies in the system are colliding then some elements of $v_\oplus$ are negative. In a system of rigid bodies an impulse must be applied to prevent the objects from inter-penetrating. While the nature of the deformations that occurs during a real collision are quite complex several analytical methods have been proposed to compute the needed impulse forces. Here we will examine one of the most common models for rigid body collision/contact. This framework, however, is sufficiently general to allow other contact models to be used.

5.1 Collision model

A common empirical model is to require that for each active contact point, the velocity after the collision must be $-\epsilon$ times the relative velocity of the contact point prior to the collision. Where ϵ is a known coefficient of restitution. This constraint can be written as:

$$v^+ = -\epsilon v^-, \tag{5}$$

where v^-, v^+ is the relative velocity vector of the contacts before and after the collision.

The above constraint only describes the behavior of the active contact points. At these the impulse force must be greater then zero,

$$p > \vec{0}. \tag{6}$$

Lastly an additional constraint is required to constrain the motion at all the contact points.

$$v_\oplus^+ \geq -\epsilon v_\oplus^-. \tag{7}$$

The active contact points satisfy this constraint by default. The constraint requires that if a contact force is inactive (contact impulse force is zero) then the relative velocity at the contact point must be at least as large as it would be if the point was active. The zero impulse force requirement on the non-active contact points can be achieved by defining

$$p_\oplus = \mathrm{S}^T p \tag{8}$$

Eq. (5–8) describe the nature of the collision; but are not in a form where the unknown impulse forces p can be solved for directly. We can rewrite the constraints, however, so that the solution no longer depends on any unknown quantities.

5.2 Finding the impulse constraint equations

The unknown impulse forces p and the contact space velocities are related by the expression:

$$p \stackrel{\text{def}}{=} \Lambda \Delta v, \tag{9}$$

where $\Delta v = v^+ - v^-$ is the change in contact space velocity caused by the collision. Combining equation 9 and 5 we can rewrite constraint equation 5 as:

$$\Lambda^{-1} p + (1+\epsilon) v^- = \vec{0}. \tag{10}$$

Observing that in equation 7 each row of the left hand side is strictly positive even if non-active points are considered and that $p > 0$ we can obtain a new form of the constraint by multiplying each side of equation 10 by p^T:

$$p^T \left(\Lambda^{-1} p + (1+\epsilon) v^-\right) = 0 \tag{11}$$

The above expression is still defined in the unknown space of the active contacts, but by noting that $v^- = \mathrm{S}v_\oplus^-$ and using the definition of Λ^{-1} (equation 4) we obtain

$$p^T \mathrm{S} \left(\Lambda_\oplus^{-1} \mathrm{S}^T p + (1+\epsilon) v_\oplus^-\right) = 0. \tag{12}$$

Finally using our definition of $p_\oplus$ from equation 8 and noting that $v_\oplus = J_\oplus q^-$ we obtain an expression for constraint equation 5 that is in a form that does not depend on any unknown parameters except $p_\oplus$:

$$p_\oplus^T \left(\Lambda_\oplus^{-1} p_\oplus + (1+\epsilon) J_\oplus q^-\right) = 0. \tag{13}$$

Eq. (5) is only valid if constraint Eq. (7) is also satisfied. Eq. (7) places an inequality constraint on all the contact points, not just the points in the active set. Only the active contact points, where the impulse forces are positive, alter the motion of the system. The change in contact space velocities caused by the active contacts can be found to be:

$$\Delta v_{\oplus} = J_{\oplus} A^{-1} J^T p. \tag{14}$$

Rewriting equation 7 using equation 14 we obtain:

$$\Lambda_{\oplus}^{-1} p_{\oplus} \geq -(1+\epsilon) J_{\oplus} q^{-}. \tag{15}$$

We can now write all the constraint equations 13,6 and 15 in a form suitable for finding the unknown impulse vector $p_{\oplus}$

Impulse Constraint Equations

$$\begin{aligned} p_{\oplus}^T \Lambda_{\oplus}^{-1} p_{\oplus} + (1+\epsilon) p_{\oplus}^T J_{\oplus} q^{-} &= 0, \\ p_{\oplus} &\geq 0, \\ \Lambda_{\oplus}^{-1} p_{\oplus} &\geq -(1+\epsilon) J_{\oplus} q^{-}. \end{aligned} \tag{16}$$

Given that $\Lambda_{\oplus}^{-1}$ is positive semi-definite such a system can be solved using a quadratic programming package such as is described by Gill [4]. In addition it should be noted that these constraints have the same form as the linear complementary problem (LCP) solved by Baraff in [1] for simple rigid body simulation. As can now be seen, the constraints described above form the same set of constraints for generalized articulated body systems as was derived for simple rigid bodies previously [2, 1].

Once a solution of the augmented contact space impulse vector $p_{\oplus}$ has been found the vector of active contact parameters and the selection matrix S can easily be computed. The non-zero terms of $p_{\oplus}$ form the active set of contact points p. The post collision in configuration space velocities $\dot{q}^{+}$ created by a contact space impulse p is given by

$$\dot{q}^{+} = \dot{q}^{-} + \Delta\dot{q} = \dot{q}^{-} + A^{-1} J^T p \tag{17}$$

and the integration of the equations of motion continued from this updated state.

5.3 Collision analysis

The largest benefit of describing the constraint equations in contact space is that the interaction between groups of dynamic systems can be described easily without having to examine the complex equations of motion of each individual system. As such, a collision model can be developed with the same ease as if one was considering interaction only between simple bodies.

Further insight into the nature of the collision constraints can be found by explicitly inverting the inverse active contact space inertia matrix Λ^{-1}. While not strictly required for the purpose of simulation but provides additional understanding about of the constraint equations. Given Λ the vector of contact space impulses p can be expressed from equation 10 as:

$$p = -(1+\epsilon)\Lambda J \dot{q}^{-}. \tag{18}$$

Inserting p into equation 17 a linear expression for $\dot{q}^{+}$ from $\dot{q}^{-}$ is obtained:

$$\dot{q}^{+} = \left[I - (1+\epsilon)\overline{J}J\right]\dot{q}^{-}, \tag{19}$$

where $\overline{J} = A^{-1} J^T \Lambda$ is the dynamically consistent generalized inverse of J. This is the unique generalized inverse of J that is consistent with the natural dynamics of the system [5, 6, 14].

Separating the ϵ term we obtain

$$\dot{q}^{+} = \overline{J}\left(-\epsilon J \dot{q}^{-}\right) + \left[I - \overline{J}J\right]\dot{q}^{-}. \tag{20}$$

The matrix $\left[I - \overline{J}J\right]$ is the basis of the null space of the contact space. Velocity vectors mapped through this space do not effect the contact space velocities. Lastly by noting that $-\epsilon J \dot{q}^{-} = -\epsilon v^{-} = v^{+}$ we see that

$$\dot{q}^{+} = \overline{J} v^{+} + \left[I - \overline{J}J\right]\dot{q}^{-}. \tag{21}$$

In this expression we can clearly see the effect of the collision constraints on the simulated system. The configuration space velocity after the collision is made up of two components:

- The configuration space velocities that realizes the desired empirical restitution velocity v^{+} with the minimum change in the kinetic energy of the system.
- Plus the prior configuration space velocities q^{-} mapped through the contact space null space $\left[I - \overline{J}J\right]$ eliminating any motion that would affect the velocities in the contact space.

Expression 21 is similar in form to the expression for the cartesian space control of redundant mechanisms [5]. Such a connection with control is natural since the constraint equations are in effect commanding an operational space velocity in the contact space while allowing motions in the redundant (unconstrained) directions to continue unmodified.

6 Contact Force Resolution

As was the case for computing collision impulse forces, contact forces can be computed in a similar manner. For brevity we will only highlight the derivation which follows closely the work done in the previous section. A resting or sliding contact occurs when $v_{\oplus} = 0$ and no penetration exists. If a negative acceleration ($a < 0$) exists at any of the contact parameters, however, objects in the environment may immediately begin to penetrate. To prevent this from occurring a contact force f must be applied at the contact points in order to prevent penetration.

As was the case for collision we can establish three constraints required to prevent penetration

$$\begin{aligned} a &= \vec{0}, \\ a_{\oplus} &\geq \vec{0}, \\ f &> 0. \end{aligned} \tag{22}$$

As in the situation with collision we will establish a selection matrix S_c that will be used to select the contact points that belong to the active set. Note that this set can be and is often different then the set of contacts active during collision.

From equation 1 we can obtain an expression for the active contact space acceleration

$$a = JA^{-1}[\Gamma - b - g] + \dot{J}\dot{q}. \tag{23}$$

Breaking the generalized torque vector Γ into actuator torques Γ_{joint}, and external forces $\Gamma_{ext} = J^T f$ we obtain a general expression for the contact space accelerations in terms of f

$$a = JA^{-1}J^T f + JA^{-1}[\Gamma_{joint} - b - g] + \dot{J}_c\dot{q}. \tag{24}$$

Now expanding $J = SJ_{\oplus}$ and following a similar line of action as we did in section 5 for collision we can obtain a similar set of constraints:

Resting/Sliding Contact Constraint Equations

$$\begin{aligned} f_{\oplus}^T\left(\Lambda_{\oplus}^{-1} f_{\oplus} + \dot{a}_{\oplus(free)}\right) &= 0, \\ f_{\oplus} &\geq 0, \\ \Lambda_{\oplus}^{-1} f_{\oplus} &\geq -a_{\oplus(free)}. \end{aligned} \tag{25}$$

Where $a_{\oplus(free)} = J_{\oplus}A^{-1}[\Gamma_{joint} - b - g] + \dot{J}_{\oplus}\dot{q}$ represents the contact space acceleration that would occur if no contact existed. As is the case with collision these constraints form a LCP system and can be solved by using a quadratic programming package like [4] or as was done by Baraff in [1].

Once the unknown augmented contact spaces forces $f_{\oplus}$ have been found the active forces f and selection matrix S_c can be trivially obtained. The configuration space acceleration $\ddot{q}$ that results from the application of the contact space forces is

$$\ddot{q} = A^{-1}(q)[\Gamma_{joint} + J_c^T f - b(q,\dot{q}) - g(q)]. \tag{26}$$

7 Combining Haptic and Dynamic Environments

Haptic interaction can be combined with the dynamic simulation to allow rich, intuitive interactions with virtual environments. Attaching the virtual proxy to a virtual object allows it to be used as a virtual tool which is no longer restricted to simple point or sphere. It shape and movement can be selected as appropriate for a given task. The constraints affixing the virtual proxy to the virtual tool may restrict all or only a few of the degrees of freedom of the virtual tool. This may be needed if the number of degrees of freedom of the haptic device is less then that of the virtual tool.

This framework has been used as the basis for the development of a dynamic haptic simulation system. Fig. 3 illustrates some of the environments that have been modeled with this system. Other environments including construction and underwater environments have also been modeled [13]. The simulations were tested on a Pentium 200MHz running Linux. Fig. 3(**left**) is one frame from an animation consisting of two puma560 manipulators (6 d.o.f. each) on which a rain of 60 large blocks is allowed to fall. A total of 366 d.o.f. are modeled. The simulation of 30 seconds took approximately 8 minutes to run.

In Fig. 3(**right**) a similar environment where two robotic manipulators and two rigid bodies (16 d.o.f) are modeled. Direct haptic interaction is permitted via a 3 d.o.f PHANToM haptic manipulator. The main haptic servo control loop was executed on a separate processor. The user is allowed to push and attach oneself to any of the objects in the environment and feel the force and impact created by their interaction. In this simulation real time performance was achieved, with only a slight slow down when the models in the environment became highly constrained. Such a situation occurs when a large number of simultaneous collisions exist between two or more objects in the environment.

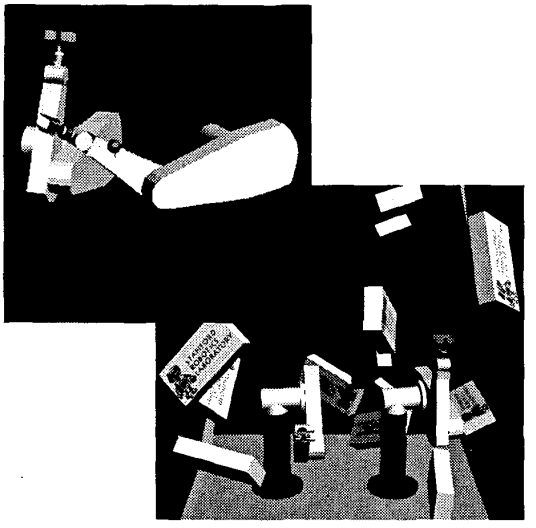

Figure 3: A frame of an animation showing the dynamic interaction of multiple articulated/rigid bodies **(right)**. A similar sequence in which direct haptic interaction is permitted between the user and the objects in the environment **(left)**.

8 Conclusion

We have presented a framework that permits the haptic interaction with complex articulated multi-body systems. The use of a generalized contact space parameters allows the interactions between arbitrarily complex models to be efficiently represented. Impact and contact forces between the bodies can then be efficiently solved to prevent penetration between all the objects in the environment. In the future we hope to apply this technology to model realistic real-world environments that cannot be easily modeled with traditional physical mock ups.

References

[1] Baraff, D., (1994), Fast Contact Force Computation for Nonpenetrating Rigid Bodies, SIGGRAPH 94 Proceedings, pp. 23-34.

[2] Baraff, D., (1989), Analytical methods for dynamic simulation of non-penetrating rigid bodies, SIGGRAPH 89 Proceedings,*Computer Graphics 23*, pp. 223-232.

[3] Featherstone, R., (1987), Robot Dynamics Algorithms, Kluwer.

[4] Gill, P., Hammarling, S., Murray, W., Saunders, M. and Wright, M., (1996), User's guide to LLSOL, Stanford University Technical Report, SOL 86-1.

[5] Khatib, O., (1990), Reduced Effective Inertia in Macro/Mini-Manipulator Systems, IEEE *Journal of Robotics and Automation*, vol. 5, num. 1, pp. 279-284.

[6] Khatib, O., (1988), Object Manipulation in a Multi-effector Robot System, IEEE *Journal of Robotics and Automation*, vol. 4, num. 1, pp. 137-144.

[7] Khatib, O., (1987), A Unified Approach to Motion and Force Control of Robot Manipulators: The Operational Space Formulation, IEEE *Journal of Robotics and Automation*, vol. 3, num. 1, pp. 43-53.

[8] Mirtich, B., (1994), Impulse-based Dynamic Simulation, Proceedings of the Workshop on the Algorithmic Foundations of Robotics.

[9] Palmer, R. S., (1987), Computational Complexity of Motion and Stability of Polygons, PhD Diss., Cornell University.

[10] Ruspini, D., Khatib, O., (1999), Collision/Contact Models for Dynamic Simulation and Haptic Interaction, 9th International Symposium of Robotics Research (ISRR'99), Snowbird, Utah, pp. 185-195.

[11] Ruspini, D., Kolarov, K., Khatib, O., (1997), The Haptic Display of Complex Graphical Environments, SIGGRAPH 97 Proceedings, pp. 345-352.

[12] Ruspini, D., Kolarov, K., Khatib, O., (1996), Graphical and Haptic Manipulation of 3D Objects, *First PHANToM User's Group Workshop*, September 27–30, Dedham, MA.

[13] Lee, P., Ruspini, D., Khatib, O., (1994), Dynamic Simulation of Interactive Robotic Environment, Proceedings of IEEE International Conference on Robotics and Automation, vol. 1, pp. 1147-1152.

[14] Russakow, J., (1995), Experiments in Manipulation and Assembly by Two-arm, Free-flying Space Robots, PhD Diss., Stanford University.

[15] C. Zilles, J. Salisbury, (1994), Constraint-based God-object Method for Haptic Display, ASME Haptic Interfaces for Virtual Environment and Teleoperator Systems, *Dynamic Systems and Control 1994*, vol. 1, pp. 146-150.