Contact Force Modeling Between Non Convex Objects Using A Nonlinear Damping Model*

Robert A LaFarge and Christopher Lewis

Intelligent Systems and Robotics Center

Sandia National Laboratories

Albuquerque, New Mexico 87185

Abstract

At Sandia National Laboratories, we are developing the ability to accurately predict motions for arbitrary numbers of bodies of arbitrary shapes experiencing multiple applied forces and intermittent contacts. In particular, we are concerned with the simulation of systems such as part feeders or mobile robots operating in realistic environments. Preliminary investigation of commercial dynamics software packages led us to the conclude that we could use a commercial code to provide everything we needed except for the contact model. We found that ADAMS best fit our needs for a simulation package. To simulate intermittent contacts, we need collision detection software that can efficiently compute the distances between non-convex objects and return the associated witness features. We also require a computationally efficient contact model for rapid simulation of impact, sustained contact under load, and transition to and from contact conditions. This paper provides a technical review of a custom hierarchical distance computation engine developed at Sandia, called the C-Space Toolkit (CSTk). In addition, we will describe an efficient contact model using a non-linear damping term developed at Ohio State. Both the CSTk and the non-linear damper have been incorporated in a simplified two-body testbed code, which is used to investigate how to correctly model the contact using these two utilities. We have incorporated this model into ADAMS SOLVER using the callable function interface. An example that illustrate the capabilities of the 9.02 release of ADAMS with our extensions is provided.

{ EMBED Word.Picture.6 }

Figure { SEQ Figure * ARABIC }: Multiple Contacts Between a Non-Convex Part and a Table Top

Introduction

Simulations of certain dynamic systems, such as part feeders and robots, almost always require modeling contact between two or more objects. For modeling part feeders, even having only one part interacting with a table top can require intricate contact modeling involving multiple contact points. Figure 1 shows a typical result from our testbed code, which uses just two-bodies (part and table) to investigate contacts. It illustrates the closest distance vectors (i.e., the vector between points on both bodies that are less than some standoff distance) computed between a non-convex part and the table top. Likewise, the simulation between a robot and its environment can get quite complicated. Sometimes the environment is well known, such as in a workcell, and sometimes the environment is very arbitrary, like that which a hopping robot might encounter while traversing a mountainous terrain. For a robot in a workcell, even having only one other object in the workspace can tax the capabilities on existing commercial simulation software to model the contact between the object and the robot (e.g., ADAMS 9.02 only models contacts between convex bodies while ADAMS 9.04 does not have contact modeling at all). For a hopping robot, the ability to characterize how the robot will react to different types of terrain (mud, high grass, steep hills, big rocks) does not exist. Both of these examples require the ability to model multiple contacts between arbitrarily shaped bodies and a highly robust contact model. Our goal is to develop contact modeling capability that is reasonably efficient yet captures a wide range of physical phenomenon from simple impact to elastic wedging and viscous effects.

* This work was supported by the United States Department of Energy under Contract DE-ACO4-94AL85000. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy.

Toward this goal, two key technical issues must be resolved. First, collision detection software that can efficiently return the distances and the witness features between bodies of arbitrary shapes must exist. Recent advances in computational geometry by Xavier [**{** REF _Ref419101126 \n **}**] and others allow for efficient collision detection and distance computation between non-convex polyhedron. At Sandia, Xavier is implementing, in configuration space (C-space), a hierarchical geometric representation combined with an algorithm for fast distance computation, collision detection, and c-space point-classification. This implementation is called the C-Space Toolkit (CSTk) and is available as a C++ library.

The second technical issue is developing a computationally efficient contact model with friction for rapid simulation of impact, sustained contact under load, and transition to and from contact conditions and allowing for multiple contacts to be simulated concurrently. There are two approaches for estimating contact forces. One is the hard contact approach, investigated by Trinkle [**{** REF _Ref419101051 \n **}**], Mirtich [**{** REF _Ref419101079 \n **}**], and others, in which the bodies are assumed to be rigid. Although appealing from an efficiency standpoint, modeling bodies as strictly rigid and persistent contacts as algebraic constraints sometimes fails to yield sufficient physical fidelity. It is known that under the assumption of Coulomb friction, the rigid body model can result in non-existent, or multiple governing equations [**{** REF _Ref419101481 \n **}**]. Furthermore, this approach does not capture the combined elastic and frictional interaction necessary to model a wedged body such as in a press fit or a Morris taper.

We have decided to use a soft contact approach, investigated by Goyal [**{** REF _Ref419100962 \n **}**], Marhefka [**{** REF _Ref419100991 \n **}**], and others, in contacts are modeled by a spring-damper. The major drawback of this approach is that it introduces stiffness into the governing equations. We have implemented the non-linear version of [**{** REF _Ref419100991 \n **}**] because it offers several advantages. First, the model produces a continuous force at the initial penetration point (it computes a zero contact force with zero penetration). The numerical integrator is not restarted when a contact is encountered. Therefore, we do not lose numerical efficiency. Second, the model includes the proper variation of the coefficient of restitution with impact velocity over a wide range of impact velocities. Another advantage is that once friction is implemented, it will allow us to investigate certain viscous effects, e.g., modeling interaction with mud.

C-Space Toolkit

A robot's configuration space (c-space) is the space of its kinematic degrees of freedom, e.g., the joint-space of an arm. Sets in c-space can be defined so as to characterize a variety of spatial relationships, such as contact between the robot and its environment. C-space techniques have been fundamental to recent basic progress in robotics areas such as motion planning and physically-based reasoning. However, practical progress has been slowed by the difficulty of implementing the c-space abstraction inside each of these applications. The Configuration Space Toolkit (CSTk) is a set of high-performance algorithms and data structures developed to meet these needs. Of primary interest to this paper is the robust collision detection provided by the CSTk.

Collision detection [**{** REF _Ref419172187 \n **}**] is a basic problem in robotics and related areas, arising in motion planning [**{** REF _Ref419172155 \n **}**], control, graphical programming, motion-preview, virtual reality, and dynamic simulation. The collision detection problem asks whether a rigid body moving along a given path intersects with any of a set of obstacles at any point on that path. In a fuller version of the problem, all contacts must also be determined. In both cases, accuracy is of extreme importance when the results of collision detection between modeled objects affect the behavior of physical robots or influence the outcomes of physical simulations, such as those used in process and product design and evaluation.

Most current methods for collision detection rely on *interference detection and/or distance computation* [**{** REF _Ref419613771 \n **}**]. Simple use of interference detection can obviously miss collisions as in Figure 2. Between queries, the triangle moves linearly from A to A'. Because there is no interference with the obstacle at the query times, simple interference checking fails to detect the collision with the obstacle. Basic distance queries at the query points would indicate that it might be possible that there is a collision, but further computations would be needed to decide. The swept hull of the moving triangle intersects the obstacle, so a swept-body interference check would detect the collision. CSTk provides a combination of basic and swept-body distance/interference calculation suitable for fast, accurate collision detection.

{ EMBED Word.Picture.6 }

Figure { SEQ Figure * ARABIC }: Linear Motion of Triangle from A to A'

For the basic distance/interference calculation, CSTk uses a hierarchical distance computation that uses Gilbert's Distance Algorithm [**{** REF _Ref419613143 \n **}**]. The boundary of a body is represented in the CSTk with a bounding-volume hierarchy generated with the algorithm described in [**{** REF _Ref419182014 \n **}**]. The hierarchy is a binary tree whose nodes each contain a convex polygon or convex polyhedron. The subtree rooted at a node represents the union of the primitives at its leaves. Thus, each node of our hierarchical geometric representation contains a conservative approximation, or wrapper, of the object represented by its subtree. In particular, our trees contain a convex hull (polyhedron) at each interior node, and a convex polygon or convex polyhedron at each leaf. The use of convex polygons and polyhedron at the nodes and leaves is required by Gilbert's algorithm.

Methods that extend the basic hierarchical distance computation to swept body distance computation, both in linear translational and combined translational and rotational sweeps have been implemented in CSTk. The methods are exact for the translational case, and include an improved conservative approximation in the rotational case. The methods are fully described in [**{** REF _Ref419101615 \n **}**] which presents simple experiments for the linear-translational case comparing the swept-body and basic techniques in distance computation, interference detection, and collision detection. These experiments indicate that computing linear-translational swept-body distance is no more than 50 percent more expensive than the basic technique in practice, and that the methods hold the potential to speed up robust collision detection.

Contact Model

The rigid body approximation is often used to model the interaction of mechanisms; however this approximation when combined with a Coulomb friction model can fail mathematically. In particular, the state derivatives can become indeterminate. For instance, consider any two surfaces in contact. Three distinct types of interaction can take place. First, they may stick together, in which case the relative velocity is zero, the frictional force is less than or equal to the coefficient of friction times the normal force and the normal acceleration is zero. Second, they may be breaking contact in which case there is no normal nor tangential force and the acceleration must be greater than zero. Finally, the surfaces may be sliding in which case the tangential force is exactly equal to the normal force times the coefficient of friction and the normal acceleration is zero. To compute the state derivatives one must first determine which of the three possible modes of motion is taking place in order to determine the algebraic constraint equations. Unfortunately, sometimes more than one mode of contact yields a consistent set of state derivatives. Worse, sometimes none of the modes yield a consistent set. Still worse, Dupont [**{** REF _Ref419101481 \n **}**], has shown that even when only one mode yields a consistent set of state derivatives, the forces that maintain the constraints can be unstable. Instability indicates that the rigid body assumption with Coulomb friction is invalid. If the circumstances that lead to indeterminate contact conditions were rare one could probably ignore this kind of mathematical irregularity. Unfortunately, this is not the case. Even the case of a single slender rod in frictional contact with a floor can produce indeterminate state derivatives.

Compliant contact models avoid the indeterminacy problem at the expense of numerical stiffness, and sometimes the expense of addition state variables. Our approach is similar to that of [**{** REF _Ref419100962 \n **}**, **{** REF _Ref419100991 \n **}**] and others that use lumped spring-damper systems to model the surface compliance. In our model, the normal force (F_N) due to a local deformation and contact damping is given by

{ EMBED Equation.2 **}**

{ EQ }{ EQ }where k is the spring constant, δ is the local deformation, and λ is the damping coefficient. The constant n is dependent on the geometry of the impacting bodies. Hertzian theory for contacting spheres suggest n = 1.5, for impacting planes n = 1. For simplicity, and since our objects are modeled as polyhedron we will use n = 1. The main effect of the non-linearity in the model is that the damping is dependent on the depth of penetration. The contact force is continuous at initial contact. Another advantage of non-linear damping over a linear damping model is that unrealistic surface sticking forces are reduced. They only occur when external forces separate the objects.

A somewhat more involved yet equally important feature of this contact model is that it accurately reflects the variation of the coefficient of restitution (e) due to impact velocity (v_i). It has been shown that at low impact velocities and for most materials with a linear elastic range, the coefficient of restitution can be approximated by the equation:

{ EMBED Equation.2 }

For sufficiently small α and v_i the coefficient of restitution can be related to the damping coefficient by:

{ EMBED Equation.2 }

More details of the implications using the above equations can be found in [{ REF _Ref419100991 \n }]. Currently, we have only implemented normal force without friction, but we are close to implementing a friction model.

Implementation

Before ADAMS was picked to be the "front end" of the our simulation effort, a small testbed code was written in C++ as a means to implement the interaction between the CSTk and the lumped spring-damper. A MovingBody class was created that contains all the information to completely model each part. The information includes the mass properties, the geometry (faceted data), the spring-damper constants (α and k), the body's standoff distance (d_s), and the equations of motion. The testbed executes like a standard simulation code except that a distance function is called during each derivative evaluation. The distance function uses the CSTk to determine if there are any witness pairs (pairs of points, one from the part and one from the table) whose distance between them (d_w) is less than d_s. A contact (C_i) is placed on the body part of each witness pair. Figure 3 shows (in 2-D) how the CSTk would find two witness pairs whose distance is less than d_s. Two contact C_0 and C_1 would be created. Because the table top is fixed in inertial space, the normal force is always in the inertial Z direction. The spring deformation at each contact (δ_i) is defined to be

{ EMBED Equation.2 }

The velocity of the contact point is the z-component of the relative velocity of the contact point to the body's cg (r x ω) added to the z-component velocity of the body's cg . The normal force equation is used to generate a body external force for each contact. An external moment is also generated. We feel that allowing for multiple contacts adds smoothing to both the applied force and the applied torque since the contact force is not coming from only one direction. The testbed provides reasonable simulation until the round-off error becomes dominate, which is usually after a couple of bounces.

{ EMBED Word.Picture.6 }

Figure { SEQ Figure * ARABIC }: Testbed Standoff and Witness Pairs Distances

The testbed allowed us to quickly implement a C++ program that uses the CSTk library and the ADAMS callable routines. To use the program, one can generate a model using ADAMS VIEW. Contact constraints are used to identify which objects are to be considered for collisions. Then, the model is exported as an ADAMS

SOLVER model. In addition to the SOLVER model, several other files are automatically written that describe the geometry of the objects. The callable program first parses the solver data set and creates a new data set having no collision command. Each part in the original collision command now has a GFORCE statement and an associated floating marker.

When the simulation is run using the callable ADAMS subroutines, the GFOSUB function is called for each object that may collide. A custom GFOSUB function first obtains the states of the bodies, then updates its own internal geometric representation of the objects. If the bodies are in close proximity to one another, a non-zero force is returned based on the contact model. Although the contact model is stiff, it is continuous, and therefore, the integrator can handle it.

Example

An example illustrating the ability of our contact detection and modeling technique is inspired by the Fisher Price Rock-N-Stack toy. This toy has a number of plastic doughnuts that fit on a cone. This example is a more visually appealing example of the classical peg-in-hole problem. Furthermore, we hope to demonstrate wedging effects when we implement the frictional contact models. Both the doughnut and the cone are simple primitives from AVIEW. When contact constraints are added under AVIEW 9.02, the polygonal models of the parts are exported along with the ADAMS model file. These geometric models are highly faceted polygonal models of the primitives as seen in Figure 4. In Figure 4, the standoff distance was made larger than ordinarily be the case so the distance vectors could be highlighted (second and third frame).

For our simulation, the cone is fixed to ground and the doughnut is dropped from above. This simple simulation was executed for sufficient time to allow the doughnut to settle onto the cylinder. In the simulation, the doughnut bounces around several times as it settles. Rotations about the axis of the cone are clearly present as the doughnut nestles down on the cone. Rotational torques are due to opposing facets on the inside of the doughnut lining up with those of the cone. In Figure 4, the proximity edges are displayed as the trajectory evolves. In the first frame, the doughnut is beyond the distance threshold, and no edges are displayed. In the second frame, several lines are added that denote the closest points between two polygons that are within the threshold distance. These lines are displayed for illustrative purposes, and no forces are present until a much smaller contact distance threshold is reached. In the third frame, the number of polygons within the threshold has increased, and the lines now form a web that will later support contact forces. The final frame, shows the doughnut after it has settled onto the cone. At this point the doughnut is being supported on the cone by a web of contact forces. At this point, the web consist of over 1000 distinct non-linear spring damper forces. These forces arise from 1500 facets that lie within the contact distance threshold. Approximately 500 of the proximity pairs are duplicates. This duplication is due to the closest point between two adjacent facets being a vertex. The corner of a cube contacting a plane has a single vertex for three faces.

{ EMBED Word.Picture.6 }

Figure { SEQ Figure * ARABIC }: Rock-N-Stack Contact Analysis Example

The vertical motion of the doughnut and it's g-force are shown in Figure 5. Several bounces are clearly present until the motion is damped out and contact is persistently maintained. The g-force will eventually settle to the doughnut weight. From Figure 5, it is clear that each impact has a finite time duration. This duration depends on the spring stiffness.

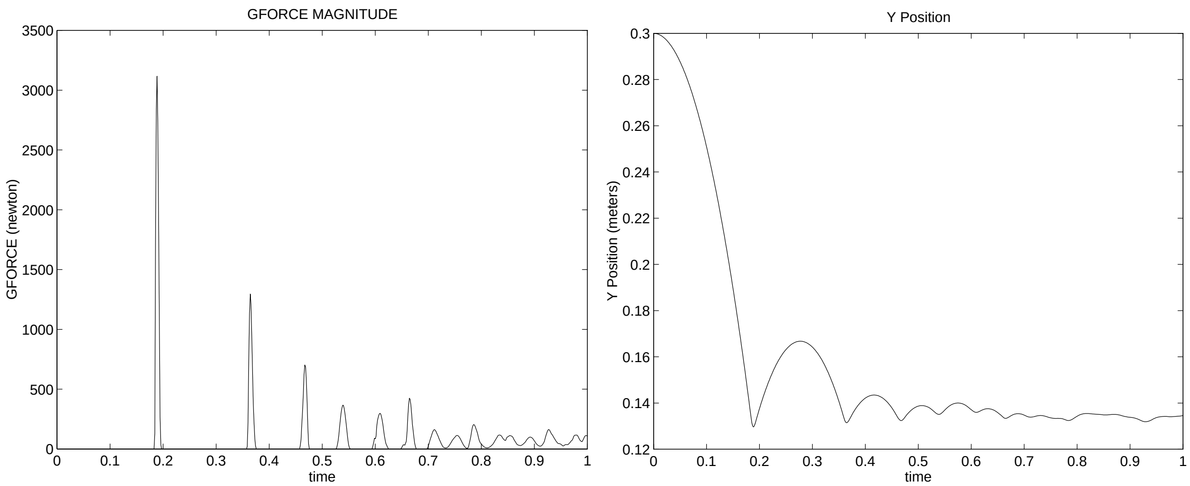

Figure { SEQ Figure * ARABIC }: Force and Position Results of Rack-N-Stack Simulation

A simple home trial with the Fisher Price toy would probably result in the falling doughnut wedging on the cone. This wedging is due to the development of relatively large normal forces caused by internal stresses that in turn cause large frictional forces. These internal stresses are necessary to the proper function of a number of common devices. A Morris taper is used on machinery usually to hold a collet in place. This simple device consists of a cone with a specific taper, and a tapered hole. Unfortunately, neither constraint-based nor impulse-based simulation are currently capable of capturing this type of physical phenomenon. It is believed however that multiple contact point soft contact models will work well once friction is implemented.

References

[1] S. Cameron, Dealing With Geometric Complexity in Motion Planning, in *Proc. 1996 IEEE ICRA Workshop on Practical Motion Planning in Robotics*, April, Minneapolis, MN.

[2] J .Cohen , M. Lin, D. Manocha, and M. Ponamgi, I-COLLIDE: An Interactive and Exact Collision Detect System for Large-Scale Environments, in *Proc. 1995 ACM Interactive 3D Graphics Conf.*

[3] E. Dupont and S. P. Yamajako, Stability of Frictional Contact in Constrained Rigid-Body Dynamics, in *Proc. 1997 IEEE Int'l Conf on Robotics and Automation*, Albuquerque, NM.

[4] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, A Fast Procedure for Computing the Distance Between Complex Objects in Three-Dimensional Space, *IEEE Journal of Robotics and Automation*, 4(2), April, 1988.

[5] S. Gottschalk, M. C. Lin, and D. Manocha, OBB-Tree: A Hierarchical Structure for Rapid Interference Detection, in *Proc. ACM SIGGRAPH '96*.

[6] Goyal, E. F. Pinson, and F. W. Sinden, Simulation of Interacting Rigid Bodies Including Friction I: General Problem and Contact Model, *Engineering With Computers*, Springer-Verlag, London, 1994.

[7] W. Marhefka and D. E. Orin, Simulation of Contact Using a Nonlinear Damping Model, *in Proc. 1997 IEEE Int'l Conf on Robotics and Automation*, Albuquerque, NM.

[8] Mirtich and J Canny, Impulse-based Simulation of Rigid Bodies, In *Proc. Monterey Symp. on Real-Time Interactive Graphics,* April 1995.

[9] C. Trinkle and J. Pang, Dynamic Multi-Rigid-Body Systems with Concurrent Distributed Contacts, in *Proc. 1997 IEEE Int'l Conf on Robotics and Automation*, Albuquerque, NM.

[10] P. G. Xavier, A Generic Algorithm for Constructing Hierarchical Representations of Geometric Objects, in *Proc. 1996 IEEE Int'l Conf. on Robotics and Automation*, April, Minneapolis, MN.

[11] P. G. Xavier, Fast Swept-Volume Distance for Robust Collision Detection, *in Proc. 1997 IEEE Int'l Conf. on Robotics and Automation*, Albuquerque, NM.

[12] P. G. Xavier and R. A. LaFarge, *A Configuration Space Toolkit for Automated Spatial Reasoning: Technical Results and LDRD Project Final Report*, SAND97-0366, February 1997, Sandia National Laboratories, Albuquerque, NM.

Filename: pap_sandia.doc
Directory: \\Davinci\UserConf
Template: C:\Program Files\Microsoft Office\Templates\Normal.dot
Title: Modeling Contact Forces On Non Convex Objects Using A Nonlinear Damping System
Subject:
Author: Robert A. LaFarge
Keywords:
Comments:
Creation Date: 06/25/98 2:02 PM
Change Number: 2
Last Saved On: 06/25/98 2:02 PM
Last Saved By: Marilyn Lee
Total Editing Time: 0 Minutes
Last Printed On: 08/06/98 3:27 PM
As of Last Complete Printing
Number of Pages: 7
Number of Words: 3,361 (approx.)
Number of Characters: 19,161 (approx.)