A Robust and Tractable Contact Model for Dynamic Robotic Simulation

Evan Drumwright
Computer Science Department
University of Memphis
edrmwrgh@memphis.edu

Dylan A. Shell
Computer Science Department
University of Southern California
dshell@usc.edu

ABSTRACT

Existing contact modeling in rigid body simulation is inadequate for robotics: no algorithms guarantee both convergence and nonpenetration at multiple contact points in the presence of Coulomb friction. We present a convex optimization based algorithm that models simultaneous contact at multiple points, ensures nonpenetration, and yields Coulomb friction effects. An example of simulated robotic grasping shows that the proposed algorithm is robust where most other methods fail.

1. INTRODUCTION

Efficient, robust algorithms for simulating robot dynamics have existed for decades. Despite being adept at simulating free space robotic movements, modeling robots interacting with objects in environments has proven to be problematic. This difficulty is partially due to the computational demands of computing contact forces that both prevent interpenetration and adhere to the Coulomb friction model. The more serious difficulty is with the limitations of the contact models for rigid body dynamics; in *inconsistent* configurations, no correct, non-impulsive contact forces exist [21]. As a result, roboticists lack the tools to simulate robots performing most tasks, even if those tasks can be modeled effectively by rigid body dynamics; this situation is unaffected by the amount of computational power available.

We present a robust approach for modeling contact that confers several advantages over the state of the art in rigid body simulation for robotics: our method handles multibodies in both maximal and generalized coordinate formulations, models simultaneous contacts without interpenetration, does not suffer from inconsistent configurations, and dispenses with threshold velocities between resting and impacting contact and sticking and sliding friction. We model contact as a convex optimization problem, for which numerically robust and fast solution methods exist.

2. BACKGROUND

SAC'09 March 8–12, 2009, Honolulu, Hawaii, U.S.A.

2.1 Penalty methods for contact

The earliest rigid body contact modeling methods were penalty methods [20]; a virtual spring and damper applies a repulsive force. Roboticists use penalty methods because they are simple to implement, especially compared to LCP methods, and very fast: e.g., near $O(1)$ running-time for convex geometries [19]. Theoretically, penalty methods achieve perfect contact modeling in the limit of zero step-size [8].

Extensive tuning of the penalty gains is required to find an acceptable tradeoff between interpenetration and numerically stiff differential equations [2]. Modeling friction with the penalty methods alters the acceleration in the direction of the contact normal, thereby increasing the complexity of tuning penalty gains significantly.

2.2 Force/acceleration LCP-based method

One popular method for determining contact forces formulates the problem as a *linear complementarity problem* (LCP) [4]. Determining contact forces for multiple points simultaneously requires counteraction of the fact that a force applied to one contact point can affect the acceleration at other points; though this consideration appears to call for the solution to a system of linear equations, *complementarity constraints* are required to ensure that forces are applied in one direction only (i.e., pushing, not pulling). This linear complementarity problem can be solved in polynomial time provided the contact forces are modeled without friction [10]. With friction the problem becomes NP-hard [3].

Perhaps more troubling than the computational requirements is the issue of inconsistent configurations, exemplified by the problem of Painlevé [21]. Counterintiuitively, the Coulomb friction model coupled with nonpenetration constraints can result in interpenetration with increasing normal forces; as Stewart [27] notes, the solution to this problem requires impulsive forces.

Notwithstanding the issues described above, one of the more popular dynamic robotics simulators, ODE [25], uses a force/acceleration LCP-based contact model (adapted from [4]). A pivoting LCP solver is augmented with a mechanism to limit variables between bounds; unlike [4], however, all friction is modeled as sticking with a friction pyramid in place of the accepted friction cone[1]. Even with these simplifications, the LCP solver is not guaranteed to terminate, though convergence is typical if the LCP matrix is well-conditioned. Matrix regularization can ensure termination with the potential cost of slight constraint violation, i.e., interpenetration.

[1]The solver determines forces to yield zero tangential ac-

2.3 Impulse-based simulation

Impulse-based simulation, introduced by Hahn [14], treats all contacts—both resting and impacting—using impulses; with this approach, nonpenetration can be guaranteed for single-point contacts, and there is no discontinuity introduced by using separate methods to treat resting and impacting contact. Impulse-based simulation was later extended to be energetically dissipative [18] and handle simultaneous contact [2, 13].

Impulse-based simulation has traditionally had difficulty modeling stable and simultaneous contacts; both are important not only for robotics applications (e.g., grasping), but general rigid-body simulation as well. For example, without effective stable contact, the ubiquitous example of stacked boxes will fail: the boxes will collapse.

2.4 Time-stepping formulations

A number of approaches toward rigid body simulation have investigated time-stepping formulations (e.g., [1, 26]) in order to address the problem of inconsistent configurations; these methods can avoid such configurations provably. These approaches consider the equations for the integrator and nonpenetration, joint, and frictional constraints simultaneously using differential inclusions [27]; a mixed-LCP (i.e., both constrained and unconstrained variables) must be solved at every step. Aside from demanding considerable computation to solve the mixed-LCP, these methods generally require multiple conditions to prove convergence to a solution (e.g., independent contact points, polyhedral geometries, etc.) Finally, time stepping methods generally use first-order, implicit Euler integrators; extension to higher order and different integration algorithms may be feasible but remains unattained.

2.5 Impact models for rigid body dynamics

Many impact models have been devised, including that of Routh [23, 5], Whittaker [29], Keller [16], Stronge [28], Smith [24], Pfeiffer and Glocker [22], and Chatterjee and Ruina [9]. These models typically are evaluated with respect to a small number of benchmark metrics, including energy consistency and ability to predict spin reversal for impacting spheres. The primary focus of impact models has remained on single point collisions. Simultaneous impacts at multiple contact points is poorly understood [8], though some models have been devised for this scenario (e.g., Pfeiffer and Glocker [22], Johansson [15]). As yet, these models do not guarantee absence of inconsistent configurations and interpenetration for multiple contact points with friction.

2.6 Introduced method vs. existing work

The table at the top of the next page compares the convex optimization impulse-based method introduced in this paper to other methods for modeling contacts in rigid body simulation.

3. CONVEX-OPTIMIZATION METHOD

Our method for modeling contact is impulse-based: both resting contact and collision are handled using impulsive forces. The Newton restitution model is used to model all contacts: the post-contact normal velocity is equal to $-\epsilon$ times the pre-contact normal velocity (for $0 \leq \epsilon \leq 1$). The first phase of the algorithm determines a set of normal impulses that conform to this model. In the second phase of the method, convex optimization is used to modify the full set of impulses (i.e., both normal and tangential); the optimization computes a set of impulses that minimizes the post-contact relative tangential velocity while keeping the post-contact relative normal velocity constant. We note that sticking and slipping contact are not handled as separate cases with this method: frictional impulses are applied to reduce tangential velocity.

3.1 Phase 1: solve for normal impulses

The first phase attempts to find a vector of impulses $\boldsymbol{x}$ such that $\mathbf{A}\boldsymbol{x} = -\boldsymbol{b}$, where $\mathbf{A}$ and $\boldsymbol{b}$ are $3n \times 3n$ and $3n \times 1$ matrices (n contact points times three basis vectors per contact point) determined using the test force method of Kokkevis [17] (with units of impulses and relative velocities in place of forces and accelerations). We denote elements in the normal and tangential coordinates by the subscripts N and T, respectively. Our method requires that $\boldsymbol{b}_{N_i} \geq 0$ for $i = 1, \ldots, n$; informally, this means that the bodies are approaching or at rest in the normal direction of every contact point.

The first phase of the algorithm begins by partitioning $\mathbf{A}$ and $\boldsymbol{b}$ into normal and tangential components:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_N & \mathbf{B} \\ \mathbf{B}^{\mathsf{T}} & \mathbf{A}_T \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_N \\ \boldsymbol{b}_T \end{bmatrix}.$$

Next, a set of impulses that drives the normal relative velocities to zero is computed using a singularity-robust inverse (e.g., SVD-based pseudo-inverse). Thus, a solution $\boldsymbol{w}_s^-$ to the following linear system of equations is sought:

$$\mathbf{A}_N \boldsymbol{w}_s^- = -\boldsymbol{b}_N \tag{1}$$

$\mathbf{A}_N$ may be ill-conditioned as a result of redundant contact points; in such cases, the SVD-based pseudo-inverse will yield the solution $\boldsymbol{w}_s^-$ with the minimum norm.

If the vector $\boldsymbol{w}_s^-$ is scaled by $(1 + \epsilon)$, the vector of impulse magnitudes is consistent with the Newton impact law without friction; this vector will be denoted $\boldsymbol{w}_s$. Augmenting $\boldsymbol{w}_s$ with $2n$ zeros creates a new vector, $\boldsymbol{w}$, which will be referenced in the following section.

3.2 Phase 2: minimize tangential velocities

The first phase determined a set of impulses magnitudes, applied in the normal directions at the contact points, that would satisfy the Newton restitution model; the effect of $\boldsymbol{w}$ on the tangential velocities was not considered. The second phase alters this set of impulse magnitudes such that the norm of the post-contact tangential velocities is minimized. This minimization is conducted over the nullspace of $\mathbf{A}_N$ to maintain the post-impulse velocities predicted by the Newton model in the normal directions.

The minimization is constrained in ways additional to the nullspace condition described above. The impulses applied in the tangential directions at a contact point are limited according to the Coulomb friction model. Additionally,

celeration. Additionally, pre-existing tangential velocity is minimized by adding a correctional term into the pre-contact tangential acceleration; this method for simulating slipping friction is inconsistent with the Coulomb friction model, but manages to circumvent the problem of Painlevé.

[2]The Stewart-Trinkle algorithm can guarantee convergence to a solution only under certain conditions.

	Time complexity	Prevents interpenetration	Convergence guaranteed	Models stable contact	Models impacts	Integrator
Penalty	$O(1)$	No	N/A	No	Yes	Any
acceleration LCP-based	$O(n^3)$	Yes	No	Yes	No	Any
Hahn impulse-based	$O(n)$	No	N/A	No	Yes	Any
Mirtich impulse-based	$O(n)$	No	N/A	No	Yes	Any
Guendelman et al. impulse-based	$O(n^3)$	Yes	Yes	No	Yes	Symplectic
Time-stepping (Stewart-Trinkle [26])	$O(n^3)$	Yes	Yes[2]	Yes	No	1^{st} order Euler
Convex optimization impulse-based	$O(n^3)$	Yes	Yes	Yes	Yes	Any

the norm of the normal components of the determined impulse magnitudes must be no greater than $||\boldsymbol{w}_s||$. This constraint prevents the normal impulse magnitudes from being increased without bound; such an increase, which would still remain within the nullspace of $\mathbf{A}_N$, can occur if $\mathbf{A}_N$ is ill-conditioned. The optimization process could then drive the normal impulse magnitudes very high, thus also permitting the tangential impulse magnitudes to be high (and the post-contact tangential velocities to be zero).

For convenience during the following discussion, we define:

$$\mathbf{C} = \left[\begin{array}{cc} \mathbf{A}_N & \mathbf{B} \end{array}\right], \qquad \mathbf{D} = \left[\begin{array}{cc} \mathbf{B}^{\mathsf{T}} & \mathbf{A}_T \end{array}\right].$$

Formally, we are searching for a vector $\boldsymbol{x}$, for which the subvector $\boldsymbol{x}_N$ satisfies $\mathbf{A}_N\boldsymbol{x}_N = -\boldsymbol{b}_N(1+\epsilon)$, and for which the norm of the vector $\mathbf{D}\boldsymbol{x} + \boldsymbol{b}_T$ is minimized. The solution $\boldsymbol{x}$ to the optimization problem can be written as:

$$\boldsymbol{x} = \boldsymbol{w} + \mathbf{R}\boldsymbol{y} \tag{2}$$

where $\mathbf{R}$ is a nullspace basis of $\mathbf{C}$ and $\boldsymbol{y}$ is the *homogeneous solution*.

Using (2) we can rewrite $\mathbf{A}\boldsymbol{x}$:

$$\mathbf{A}(\boldsymbol{w} + \mathbf{R}\boldsymbol{y}) = \left[\begin{array}{c} -\boldsymbol{b}_N(1+\epsilon) \\ \mathbf{D}(\boldsymbol{w} + \mathbf{R}\,\boldsymbol{y}) \end{array}\right].$$

Formally, the optimization problem to be solved is:

$$\begin{array}{lrcl} \textbf{Minimize} & f_0(\boldsymbol{y}) & = & \boldsymbol{y}^{\mathsf{T}}\,\mathbf{U}\boldsymbol{y} + \boldsymbol{v}^{\mathsf{T}}\boldsymbol{y} \\ \textbf{Subject to} & 0 & \geq & f_i(\boldsymbol{y}) = (\boldsymbol{y}^{\mathsf{T}}\boldsymbol{r}^{\mathsf{T}}{}_{n+2i-1})^2 + (\boldsymbol{y}^{\mathsf{T}}\boldsymbol{r}^{\mathsf{T}}{}_{n+2i})^2 \\ & & & \qquad -\mu^2\,(w_i + \boldsymbol{y}^{\mathsf{T}}\boldsymbol{r}^{\mathsf{T}}{}_i)^2 \\ & & & \text{where } i \in [1,\ldots,n], \quad \mu \in [0,\infty], \\ & & & \text{and } \mathbf{U} = \boldsymbol{R}^{\mathsf{T}}\boldsymbol{D}^{\mathsf{T}}\boldsymbol{D}\,\boldsymbol{R} \\ & & & \text{and } \boldsymbol{v}^{\mathsf{T}} = 2\,(\boldsymbol{w}^{\mathsf{T}}\boldsymbol{D} + \boldsymbol{b}_T^{\mathsf{T}})\boldsymbol{D}\boldsymbol{R} \\ & 0 & \geq & g(\boldsymbol{y}) = (2\boldsymbol{w}_s + \mathbf{S}\boldsymbol{y})^T\mathbf{S}\boldsymbol{y} \end{array}$$

Here, as well as in the derivations that follow, we use the notation $\boldsymbol{a}_i$ for the i^{th} column vector of matrix $\mathbf{A}$, and $\boldsymbol{a}^{\mathsf{T}}{}_j$ for the j^{th} row vector of matrix $\mathbf{A}$, and $a_{i,j} = (\boldsymbol{a}_i)_j = (\boldsymbol{a}^{\mathsf{T}}{}_j)_i$, the $(i^{\text{th}}, j^{\text{th}})$ element of $\mathbf{A}$.

Convexity of this optimization problem follows from the symmetry and positive semi-definiteness of $\mathbf{A}$[3] and a feasible region that is an intersection of convex sets. This convexity implies that any local solution $\boldsymbol{x}^*$ is a global solution to the optimization problem.

[3]Mirtich [18] proved that the 3×3 collision matrix, $\mathbf{K}$, is symmetric positive semi-definite; $\mathbf{A}$ is a straightforward extension of $\mathbf{K}$ from one to n contact points, and is proven to be symmetric positive semi-definite similarly.

A standard algorithm in convex optimization, such as the *Logarithmic Barrier* method [6], is able to solve the above problem, which contains both inequality and equality constraints. The Logarithmic Barrier method approximates an optimization problem with inequality constraints as an unconstrained optimization problem by transforming the inequality constraints by the logarithm function; values of $\boldsymbol{y}$ that violate the constraints drive the objective function to infinity. The Logarithmic Barrier method is ϵ-suboptimal [6], and will not permit any of the inequality constraints to be violated: the solution $\boldsymbol{x}^*$ will always be feasible. The objective function to be minimized takes the form:

$$h(\boldsymbol{y}) = f_0(\boldsymbol{y}) + \left(\frac{-1}{t}\right)\log(-g(\boldsymbol{y})) + \sum_{i=1}^{n}\left(\frac{-1}{t}\right)\log(-f_i(\boldsymbol{y}))$$

where t is a parameter that yields better approximation to the global optimum as it increases, and $f_i(\cdot) \leq 0$, for $i = 1,\ldots,n$ when all inequality constraints are satisfied. The minimization of the above problem can be found via application of *Newton's method* to find descent directions $\Delta\boldsymbol{y}$:

$$\Delta\boldsymbol{y} = -\alpha\nabla^2 h(\boldsymbol{y})^{-1}\nabla h(\boldsymbol{y})$$

where $h(\cdot)$ is the function to be minimized with respect to $\boldsymbol{y}$ and α is the value that minimizes $h(\boldsymbol{y} + \Delta\boldsymbol{y})$; α is found by line search. Given that we wish to minimize $h(\boldsymbol{y})$, the gradient and Hessian of this function are needed; they can be derived trivially.

Using the Logarithmic Barrier method along with Newton's method [6] and f_0, f_i, and g_0 (along with corresponding gradients and Hessians), the contact impulse magnitudes can be determined. We use Brent's method [7] for line search to determine the value of α that minimizes $h(\boldsymbol{y} + \Delta\boldsymbol{y})$.

3.3 Complexity analysis

The first phase of our algorithm requires the determination of the particular solution using the singular-value decomposition, which exhibits complexity of $O(n^3)$. The convex optimization in the second phase requires a solution to a system of linear equations in each Newton (inner) iteration, exhibiting complexity of $O(n^{2.376})$. The number of outer iterations performed by the Logarithmic Barrier method is more difficult to quantify; however, Boyd and Vanderberghe [6] note that a conservative estimate is that the total number of Newton iterations is bounded by $\sqrt{n}\log n$, where n is the number of constraints in the problem (i.e., identical to $n+1$ in § 3.2). Thus, the overall complexity is $O(n^3 + n^{2.876}\log n) = O(n^3)$; practical results using Cholesky decomposition (the Hessian is symmetric and positive-definite) are of order $n^{3.5}\log n$.

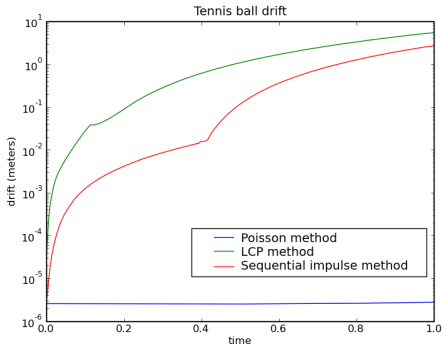

Figure 1: Plot of the tennis ball's drift from its initial position in the robot's gripper during the grasping task; the y axis is logarithmic scale. Note that, after approximately 0.16 seconds, the force/acceleration LCP-based method begins to drop the ball (evident by the discontinuity in the graph); after approximately 0.4 seconds, the sequential impulse-based method begins to drop the ball. After grip is lost, positional error begins to increase quadratically due to gravity, making further comparisons between the methods impossible.

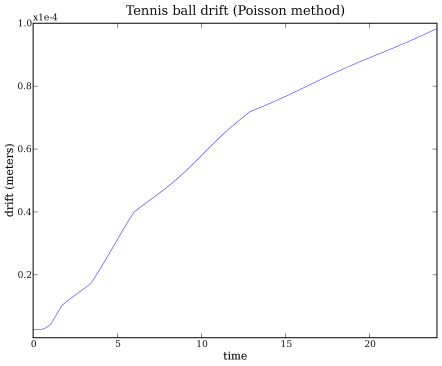

Figure 2: Plot of the tennis ball's drift from its initial position in the robot's gripper during the grasping task. The convex optimization based method was used to model contact. Drift is shown over the entire 24 seconds (contrast against Fig. 1) that it took for the robot to complete the task. Note that the y axis is linear scale.

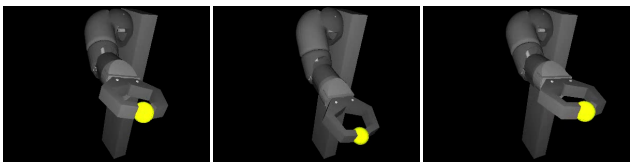

Figure 3: Three snapshots of the simulated manipulator grasping a tennis ball while following a sinusoidal trajectory.

4. EXAMPLE

Simulated robot grasping has proven particularly difficult to model with rigid body simulation for multiple reasons, including the closed-chain configuration that can result from force closure, the complexities of modeling sticking friction accurately, and the necessity of minimizing tangential drift the bodies in contact. Miller successfully simulated this task using a modified version of the time-stepping method of Anitescu and Potra [1]; however, as discussed in § 2.4, the utility of Miller's method is limited from a robotics perspective. Most significantly, the time-stepping methods use the maximal coordinate formulation, which precludes the use of advanced (i.e., inverse dynamics) control techniques and makes it impractical to obtain useful matrices like the joint-space inertia matrix and the manipulator Jacobian.

The grasping example consists of a 7 degree-of-freedom simulated manipulator setup with a tennisball positioned snugly within the robot's gripper; the goal of this example is to get the robot arm to follow a sinusoidal trajectory while grasping the tennis ball. The robot is simulated using Featherstone's articulated body method [12]; the robot is controlled using a composite proportional-derivative (PD) / feedforward (Recursive Newton-Euler inverse dynamics [12]) method. The robot's two shoulder joints are driven to follow the path $\frac{\sin(t)}{4}$ and $\frac{\sin(2t)}{4}$ over 25 seconds, which allows for four cycles of the movement. A desired acceleration of $100m/s^2$ for the gripper joints is used in combination with inverse dynamics to generate torques for grasping the ball; for simplicity, contact between the ball and the gripper is modeled as perfectly inelastic[4]. The simulation step size used was 0.001 and explicit Euler integration was employed.

The penalty method of Drumwright [11] was first tested on this example. Regardless of the gains and friction coefficient selected[5], the penalty method caused the robot to drop the ball immediately (within three simulation iterations): the robot would squeeze too tightly, not tightly enough, or the net contact forces on the ball would not be aligned with the contact normal. Additionally, sticking friction was modeled with virtual springs and dampers, which would permit movement of the ball along the gripper surface; this movement was followed by attempted correction and, generally, oscillation, which prohibited the robot from grasping the ball even if the arm was kept stationary.

The force/acceleration LCP-based method described in § 2.2 was next attempted on this example. As shown in Fig. 1, the LCP-based method managed to keep the ball grasped for fewer than two hundred simulation iterations; no coefficient of friction could be selected to keep the ball grasped longer. The LCP solver always converged successfully; however, as Fig. 1 indicates, significant tangential slip was observed. Generation of unequal actuator torques for the gripper jaws by the controller was eliminated as the cause of this problem[6]; we tried manually equalizing the actuator torques for the gripper jaws without positive effect.

[4]Theoretically, modeling the contact with elasticity should not be an issue for our method: the contacts at the opposing jaws of the gripper would eliminate any restitution in the collision. However, this scenario was not attempted.

[5]The penalty method of Drumwright [11] is able to effect the behavior of the standard penalty method [20] through parameter settings; the standard penalty method fails on this example equally quickly.

[6]All forces on the robot must be known to compute inverse dynamics, and the contact forces on the robot change as a

We posit that the constraint (i.e., nonpenetration and friction) error resulting from matrix regularization causes the task to fail so rapidly. Without a reference set of forces that should be applied for comparison, the exact cause of failure could not be determined.

The impulse-based method of Guendelman et al. [13] was tried on this example as well. This method manages to retain control of the ball roughly twice as long as the LCP-based method, which is surprising given that the former method has difficulty simulating stable contact. Analysis of the post-contact velocities of the impulse-based method indicates that it reduces the tangential velocity of the ball fairly effectively, but that excess energy generated from the contact pushes the gripper jaws apart.

The convex optimization-based method introduced in this paper was able to complete this example, as depicted in Figs 2 and 3. Fig. 2 indicates that, by the end of the movement, the drift approaches 0.1mm. We continued running this example for 100 seconds total, and the rate of drift observed from 15 and 20 seconds (see Fig. 2) continued; the robot maintained its grip on the ball throughout the 100 seconds. It is almost certain that at some point the robot would drop the ball. However, it is conceivable that the robot could readjust its grasp.

The impulse-based approach provides a side benefit in the computation of inverse dynamics: contacts are handled at the start of the simulation step, and thus no contact forces need be considered in the inverse dynamics calculations[7].

5. DISCUSSION

Our primary goal is to facilitate rigid body simulation for robotics, rather than to provide a more predictive and accurate collision law. Toward this goal, we have provided a method that is consistent and asymptotically efficient; the Logarithmic Barrier method that is used as the basis of our algorithm is well-studied and quite robust. Additionally, our method always respects nonpenetration and Coulomb friction constraints for multiple, simultaneous contact points.

Our method always converges, guarantees nonpenetration and friction constraints are met at all contact points, models stable contact, and its worst-case asymptotic performance is polynomial in the number of contacts; additional benefits provided by our method include dispensation with thresholding between resting and impacting contacts and sticking and sliding friction, and the ability to compute inverse dynamics accurately during contact. There is no existing contact model for rigid body simulation that can make all of these claims.

6. REFERENCES

[1] M. Anitescu, F. Potra, and D. Stewart. Time-stepping for three dimensional rigid body dynamics. *Computer Methods in Applied Mechanics and Engineering*, 177:183–197, 1999.

result of the motor torques applied by the controller *in silico*. This coupling between forward dynamics with Coulomb friction and inverse dynamics leads to the latter being computed only approximately correctly when the LCP-based method is used.

[7]A first-order approximation of the contact forces can be obtained, if desired, by scaling the impulses determined by the contact method by $1/h$, where h is the integration step size.

[2] D. Baraff. Analytical methods for dynamic simulation of non-penetrating rigid bodies. *Computer Graphics*, 23(3), July 1989.

[3] D. Baraff. Coping with friction for non-penetrating rigid body simulation. *Computer Graphics*, 25(4):31–40, 1991.

[4] D. Baraff. Fast contact force computation for nonpenetrating rigid bodies. In *Proc. of SIGGRAPH*, Orlando, FL, July 1994.

[5] V. Bhatt and J. Koechling. Three-dimensional frictional rigid-body impact. *ASME J. Appl. Mech.*, 62:893–896, 1995.

[6] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[7] R. P. Brent. *Algorithms for Minimization without Derivatives*. Prentice-Hall, Engelwood Cliffs, NJ, 1973.

[8] B. Brogliato, A. ten Dam, L. Paoli, F. Génot, and M. Abadie. Numerical simulation of finite dimensional multibody nonsmooth mechanical systems. *ASME Appl. Mech. Reviews*, 55(2):107–150, March 2002.

[9] A. Chatterjee and A. Ruina. A new algebraic rigid body collision law based on impulse space considerations. *ASME J. Appl. Mech.*, 65(4):939–951, Dec 1998.

[10] R. W. Cottle, J.-S. Pang, and R. Stone. *The Linear Complementarity Problem*. Academic Press, Boston, 1992.

[11] E. Drumwright. A fast and stable penalty method for rigid body simulation. *IEEE Trans. on Visualization and Computer Graphics*, 14(1):231–240, Jan/Feb 2008.

[12] R. Featherstone. *Robot Dynamics Algorithms*. Kluwer, 1987.

[13] E. Guendelman, R. Bridson, and R. Fedkiw. Nonconvex rigid bodies with stacking. *ACM Trans. on Graphics*, 22(3):871–878, 2003.

[14] J. K. Hahn. Realistic animation of rigid bodies. *Computer Graphics*, 22(4), 1988.

[15] L. Johansson. A newton method for rigid body frictional impact with multiple simultaneous impact points. *Comput. Methods Appl. Mech. Engrg.*, 191:239–254, 2001.

[16] J. B. Keller. Impact with friction. *ASME J. Appl. Mech.*, 53:1–3, Mar 1986.

[17] E. Kokkevis. Practical physics for articulated characters. In *Proc. of Game Developers Conf.*, 2004.

[18] B. Mirtich. *Impulse-based Dynamic Simulation of Rigid Body Systems*. PhD thesis, University of California, Berkeley, 1996.

[19] B. Mirtich. V-Clip: fast and robust polyhedral collision detection. *ACM Trans. on Graphics*, 17(3):177–208, 1998.

[20] M. Moore and J. Wilhelms. Collision detection and response for computer animation. In *Proc. of Intl. Conf. on Computer Graphics and Interactive Techniques*, pages 289–298, 1988.

[21] P. Painlevé. Sur le lois du frottement de glissemment. *C. R. Académie des Sciences Paris*, 121:112–115, 1895.

[22] F. Pfeiffer and C. Glocker. *Dynamics of rigid body systems with unilateral constraints*. John Wiley and Sons, 1996.

[23] E. J. Routh. *Dynamics of a system of rigid bodies, 6th ed.* Macmillan and Co., London, 1897.

[24] C. E. Smith. Predicting rebounds using rigid-body dynamics. *ASME J. Appl. Mech.*, 58:754–758, 1991.

[25] R. Smith. ODE: Open Dynamics Engine.

[26] D. Stewart and J. Trinkle. An implicit time-stepping scheme for rigid body dynamics with coulomb friction. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, San Francisco, CA, April 2000.

[27] D. E. Stewart. Rigid-body dynamics with friction and impact. *SIAM Review*, 42(1):3–39, Mar 2000.

[28] W. J. Stronge. Rigid body collisions with friction. *Proc. of the Royal Society of London A*, 431(169–181), 1990.

[29] E. T. Whittaker. *A Treatise on the Analytical Dynamics of Particles and Rigid Bodies, 4th ed.* Dover, 1944.