Symbolic and Numerical Computation for Artificial Intelligence

edited by

Bruce Randall Donald
Department of Computer Science
Cornell University, USA

Deepak Kapur
Department of Computer Science
State University of New York, USA

Joseph L. Mundy
AI Laboratory
GE Corporate R&D, Schenectady, USA

Academic Press
Harcourt Brace Jovanovich, Publishers
London San Diego New York
Boston Sydney Tokyo Toronto

ACADEMIC PRESS LIMITED
24–28 Oval Road
London NW1

US edition published by
ACADEMIC PRESS INC.
San Diego, CA 92101

This book is printed on acid-free paper

A catalogue record for this book is available from the British Library

ISBN 0-12-220535-9

Printed and Bound in Great Britain by
The University Press, Cambridge

Chapter 10

Symbolic Methods for the Simulation of Planar Mechanical Systems in Design

Bruce R. Donald†

Computer Science Department

Cornell University, Ithaca, NY 14853

Dinesh K. Pai‡

Department of Computer Science

University of British Columbia, Vancouver, Canada V6T 1Z2

Certain common mechanical systems can be simulated using purely algebraic methods, requiring no numerical integration. In this approach, we reduce the simulation problem to sweeping an arrangement of plane algebraic curves of low degree, which in turn reduce to algebraic intersection problems. We provide here an introduction to the problem, the simulation approach, and applications to design for assembly.

---

1. Introduction

Simulation of mechanical systems for design, analysis, and testing has traditionally been considered amenable only by numerical methods. Symbolic methods have been used, if at all, only for a few aspects of simulation such as the preparation of equations for numerical integration. However, in certain situations the simulation problem can be radically reformulated to allow fast simulation using symbolic methods and no numerical integration. Such a situation arises in the design of snap-fasteners and other compliantly-connected mechanical parts — under certain assumptions, the simulation problem can be reduced to sweeping an arrangement of algebraic curves in the plane (Donald and Pai, 1991). Here, we outline the formulation, describe the role of symbolic methods in the formulation and the application of the approach to problems in design. More details can be found in Donald and Pai (1991).

† Supported in part by the National Science Foundation under grants No. IRI-8802390, IRI-9000532 and by a Presidential Young Investigator award No. IRI-8957316, and in part by the Mathematical Science Institute.

‡ Supported in part by ONR Grant N00014-88K-0591, ONR Grant N00014-89J-1946, and NSF Grant DMC-86-17355.

The motivation for our problem is *design for assembly* and more specifically the design of objects that require no additional fasteners such as screws. Such parts deform and "snap" together during assembly and considerably simplify their manufacture. Interestingly, the force required to assemble two parts may be much less than the force required to take them apart.

Despite the increasing use of such parts, there exist no general techniques for automating their design. In general, the simulation of such systems requires the solution of the mechanics of contact and deformation in the parts. These numerically intensive problems are prohibitively expensive for use in design methods which are interactive or perform extensive search of the design space. We choose instead to reformulate the simulation problem using symbolic methods and using a simpler but sufficiently accurate model of the interaction between the parts.

Our approach deviates significantly from earlier simulation methods. A major impediment to simulation of these systems has been the apparent necessity to integrate out the differential mechanics in order to determine the long-term behavior of the system. This problem is exacerbated by the fact that in many models of rotational compliance such as the generalized damper (e.g. Lozano-Pérez *et al.*, 1984; Erdmann, 1986; Donald, 1988, 1990; Canny, 1989), the resulting trajectories are not known to be algebraic; neither do we have ways of computing algebraic bounding approximations (or forward projections). Hence the traditional numerical approach to simulating such systems has been the following:

Typical Simulation Algorithm

1. Given a state x of the system, numerically integrate the differential equation governing motion of the system. Step forward in time to obtain (approximately) new state x'.
2. Perform collision detection either at x' or along the path from x to x'.
3. If the constraints have changed, reformulate the differential equation.
4. Repeat.

Numerical simulation of mechanical systems is fraught with error, special cases, and numerical problems. They are rarely combinatorially precise, and almost never come with guarantees of accuracy. We show how in the case of our system, numerical simulation can be avoided, and exact solutions can be obtained. We cannot claim that this can be done in general. However, our method yields, in this case, computationally efficient, exact solutions, and may possibly be useful in other domains.

We view the simulation problem, with rotational compliance and quasi-static mechanics, as a problem that can be solved by careful reduction to a plane sweep. In particular, for each pawl, we reduce to sweeping a planar arrangement of algebraic curves of low degree. The size of the connected component of the free space to sweep is small — almost linear in the number of constraints. The approach of Donald and Pai (1991) to modeling rotational compliance and to incorporating frictional constraints leads to the first formulation of the simulation problem which permits a *reduction* of motion prediction to plane sweep. Our solution differs from previous work on predicting, bounding, and planning rotationally compliant motions with quasi-static mechanics in that it is (i)

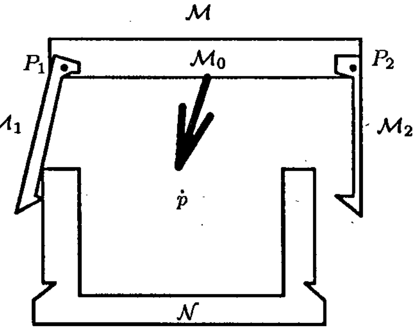

Figure 1. Linked body $\mathcal{M}$ moving among $\mathcal{N}$.

purely algebraic, and hence exact, (ii) *combinatorially precise*, in that the computational complexity is exactly known, and (iii) requires no integration.†

2. Formulation of problem

We model the interaction between a flexible part $\mathcal{M}$ and the environment $\mathcal{N}$ (including its mating parts) as follows. The flexible part $\mathcal{M}$ is constructed by attaching polygons $\mathcal{M}_h$, $h = 1, \ldots, l$, called "pawls", to a root polygon $\mathcal{M}_0$, at hinge points P_h. Each hinge is coupled with a spring of stiffness k_h. (See figure 1). The motion of the body $\mathcal{M}$ consists of rigid translation of the root polygon $\mathcal{M}_0$. The pawls $\mathcal{M}_h$ are free to move compliantly as dictated by interactions with the environment and the spring. The parts in contact obey Coulomb's law of friction and are in quasi-static motion.

The simulation problem is to determine:

1 The time of termination of the motion, the cause of termination (such as sticking due to friction, sticking due to kinematic constraints, etc.), and the configuration of $\mathcal{M}$ at termination.
2 The time history of contacts between $\mathcal{M}$ and $\mathcal{N}$.

Some extensions to this problem are considered in Donald and Pai (1991), including the determination of the time history of contact and assembly forces, and the effect of uncertainty.

We make the following assumptions about the physics of object interactions and the motion of $\mathcal{M}$:

- Object interactions are restricted to those between $\mathcal{M}_h$ and $\mathcal{N}$. In other words, pawls do not collide with each other, but may collide with the environment $\mathcal{N}$. The effect of this assumption is to make the motion of each pawl independent of the motion of other pawls. Henceforth we shall consider the motion of a single pawl $\mathcal{M}_h$.
- Since the root $\mathcal{M}_0$ is undergoing a rigid translation, so does the hinge point P_h.

† Note that Donald (1988), Canny (1989), Briggs (1989) and Friedman *et al.* (1989) address geometric reachability issues for translationally compliant objects, but these objects cannot rotate.

We shall assume that this translation is a straight line motion given by

$$p = p_0 + \dot{p}t \tag{2.1}$$

where t is the time, p_0 is the initial position (at $t = 0$) of P_h, and $\dot{p}$ is its velocity.

- Stable contact: Suppose the pawl is in contact with a feature of the environment (for example, during sliding). We assume that if we perform a small displacement of the pawl away from the environment, the torque on the pawl due to the spring is such that the contact will be restored. This assumption is not very restrictive at all – in fact, in the face of even the smallest uncertainty, stable contacts are the only ones one can hope to observe in practice.
- Quasi-static motion: The motion is assumed to be slow enough that inertial effects are not significant. This corresponds to assuming that there is no acceleration of the pawl, and hence the forces on the pawl are balanced. The quasi-static assumption is reasonable at small speeds and is widely used (see, for example, Whitney, 1982; Mason, 1982; Pai, 1988; Donald, 1990).
- If the pawl slides off $\mathcal{N}$ into free space, it may have a residual torque due to the spring being cocked. We shall assume that the pawl rotates back towards its rest orientation at such great speed that p does not change significantly during the rotation and can be taken to be constant. This, incidentally, is the "snap" in the snap-fasteners that we wish to model. Quasi-static motion implies that the root is moving "slow-enough" for the forces to be balanced. Hence, when a pawl is in free space and has a residual torque, its motion can be fast compared to that of the root, resulting in a "snap". This assumption can be relaxed by assuming a linear relationship between the translation p and the rotation (e.g. Canny, 1986), but we do not deal with it here.
- The forces of friction arising from contact obey Coulomb's Law. We further assume that there is a single coefficient of friction. This assumption is also widely used.

3. Outline of Simulation

Given the above model of flexible objects as compliantly connected rigid bodies, our task is to simulate their behavior efficiently. We proceed by transforming the problem into an appropriate configuration space, in which obstacles and other constraints are reduced to algebraic curves of low degree.

Two types of contact are possible between the pawl $\mathcal{M}_h$ and a polygon in $\mathcal{N}$. Following the convention of Lozano-Pérez (1983), Canny (1986) and Donald (1987), we say that Type-A contact occurs when a vertex of $\mathcal{N}$ touches an edge of the pawl; Type-B contact occurs when a vertex of the pawl touches an edge of $\mathcal{N}$.

We can now write the contact constraint equations for the two types of contact, as in Canny (1986). We shall index features (vertices and edges) of the moving pawl by the subscript i and features of the polygonal environment by the subscript j. Let an edge of $\mathcal{M}_h$ be represented by its outward normal, n_i, and its distance to the hinge point along the normal, d_i. Let p_j be a vector to the contact vertex of $\mathcal{N}$. Let R_θ be the linear transformation which rotates a vector by an angle θ. Then the type-A constraint can be written as

$$(p_j - p)\cdot R_\theta n_i - d_i = 0. \tag{3.1}$$

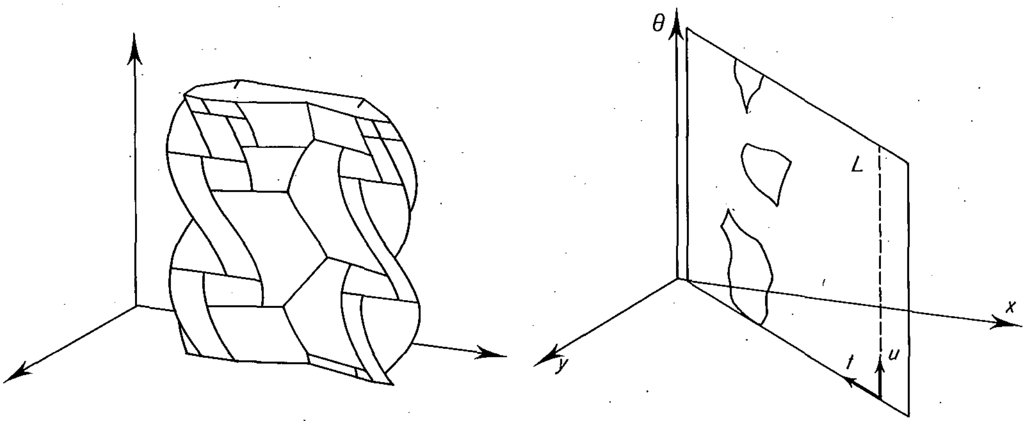

Figure 2. Configuration space constraints for a moving pawl. (Reprinted courtesy of Randy Brost (Brost, 1989). The sweep plane $L(t)$ intersects the cspace obstacles in a planar arrangement of cubic curves.

Similarly, the type-B constraint can be written as

$$(R_\theta \boldsymbol{p}_i + \boldsymbol{p})\cdot \boldsymbol{n}_j - d_j = 0, \tag{3.2}$$

where $\boldsymbol{p}_i$ is the vector from the hinge point P_h to the contact vertex of $\mathcal{M}_h$.

Consider the motion of a single pawl $\mathcal{M}_h$, whose translation is governed by (2.1). The obstacle constraints (3.1) and (3.2) can be reduced to algebraic equations by the substitution $u = \tan\frac{\theta}{2}$. This yields constraint equations that are quadratic in u, with coefficients that are affine in x and y. Since x and y are affinely parametric in time t, the coefficients are also affine in t. Intersecting two of these constraints requires intersecting two quadratics. Pure rotational intersection detection (the "snap") requires solving a quadratic in u. Pure translational intersections require solving an affine equation. Hence there exists a closed-form, purely algebraic solution to these intersection problems. The collision detection for each obstacle constraint can be done in constant time, since the degree, size, and number of variables in the constraint polynomials is fixed.

Thus the n constraints given by (3.1) and (3.2) are manifest as algebraic ruled surfaces $\{f_1, \ldots, f_n\}$ in a 3D configuration space, C, with coordinates (x, y, u). Each surface is only "applicable" for some range of orientations $[u_0, u_1]$, by which we mean that the surface only "exists" for u in this range (Donald, 1987). See figure 2 from Brost (1989).†

Let $\hat{\boldsymbol{u}}$ be a vector along the u-axis (think of $\hat{\boldsymbol{u}}$ as $(0, 0, 1)$). Now, the constraint of pure translation (2.1) of the hinge point P_h of the pawl $\mathcal{M}_h$ restricts any possible evolution of the system to lie in a 2D cylinder Y of (x, y, u)-space. We call this "plane" P_Y; it has "normal" $\hat{\boldsymbol{u}} \times (\dot{\boldsymbol{p}}, 0)$, and it corresponds to a chart for the cylinder Y.

As time t increases, the vertical line $L(t)$ sweeps across P_Y. This line contains the state of the system. It is our task to calculate the u coordinate as t evolves (i.e. increases). Now, P_Y has degree 1 and hence when intersected with a constraint f_i we obtain a cubic‡ curve

† We thank R. Brost for providing us with these figures, from Brost (1989).

‡ In fact, each curve γ_i is simultaneously quadratic in u and linear in t.

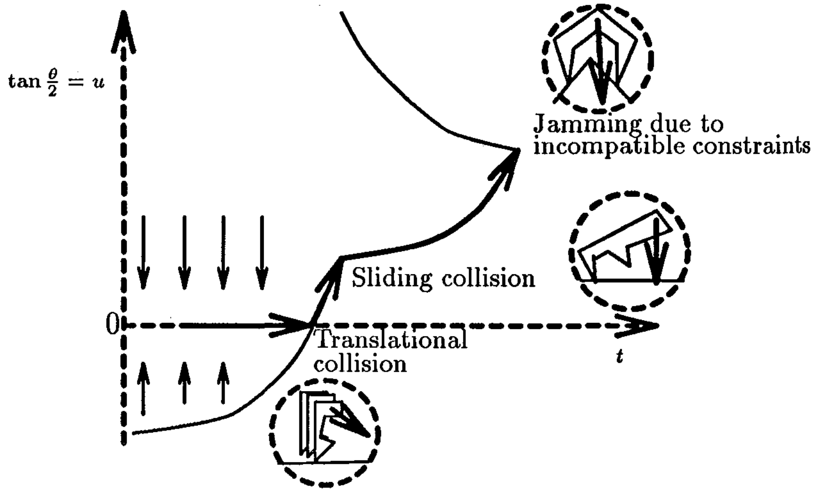

Figure 3. Sweep events: translational collision, sliding collision, and jamming due to incompatible kinematics.

segment γ_i in the (t, u) plane. So all the configuration space constraints are manifest as an arrangement of cubic curve segments $\{\gamma_1, \ldots, \gamma_n\}$ in this plane, where $\gamma_i = f_i \cap P_Y$ $(i = 1, \ldots, n)$.

In our algorithm, the sweep line sweeps across this planar arrangement of curves, and as we sweep, we compute the trajectory of the system. "Events" caused by crossing the curves γ_i will modify the trajectory. We define the following sweep events: (i) *translational collision,* (ii) *sliding collision,* (iii) *jamming due to incompatible kinematics,* (iv) *snapping free from a single constraint,* (v) *jamming on a single constraint,* (vi) *snapping free from a vertex,* and finally an event which depends on the presence of friction — (vii) a *sticking* event. Each event will have to be *handled*, by which we mean that the solution trajectory we compute may be modified. In between events, the trajectory is piecewise algebraic.

Examples of these events are depicted in figure 3. We can explain the trajectory computation algorithm like this: the dynamical system has the following geometric interpretation in slice P_Y. In P_Y, the line $u = 0$ is an *attractor*, and we imagine a vector field on P_Y parallel to the u-axis and pointing towards the t-axis. Hence the attracting vectors are parallel to $-\widehat{u}$ for $u > 0$ and parallel to $+\widehat{u}$ for $u < 0$. The curves γ_i act as (holonomic) constraints. The sweep point cannot cross these curves, but it can follow them as $L(t)$ moves. They can prevent motion of the sweep point from attaining $u = 0$.

The simulation problem is thus reduced to sweeping an arrangement of algebraic curves in the plane. The connected component of free space defined in this planar arrangement has complexity $O(\lambda_r(n))$, and can be constructed in time $O(\lambda_r(n) \log^2 n)$ using a red-blue merge algorithm (Guibas *et al.*, 1988); see Donald and Pai (1991) for more details. Here $\lambda_r(n)$ is the (almost linear) maximum length of (n, r) Davenport Schinzel sequences (Guibas *et al.*, 1988), and r is a small constant related to the number of times two cubic[†] configuration space constraint curves can intersect.

† By "cubic" we mean the total degree of the defining multinomial is 3. Our curves have additional structure, such as low-degree parameterizations, as well.

3.1. Computing Kinematic Events

Here we treat the frictionless case, defining six types of local geometric events which are purely kinematic (for example, see figure 3). If the trajectory is at the $u = 0$ position and the sweep point $z(t)$ encounters a constraint γ_i, then the sweep point complies to the constraint and is forced to move away from the zero line (u becomes positive here). This corresponds to a pure translational collision, followed by a continued motion of the root which "cocks" the pawl against an obstacle. During this motion, the sweep point follows γ_i. If a new constraint γ_j is reached, then the sweep point slides along the curve γ_j in turn. This corresponds to a sliding collision: while sliding on constraint γ_i, the pawl hits constraint γ_j. The motion continues, following γ_j compliantly. Hence the sliding collision can result in a constraint change. Finally, if the sweep point is following a curve γ_i which crosses $u = 0$, the trajectory breaks contact there and continues along the t-axis. This event is a "dual" subcase of type (i).

As can be seen from figure 3, some constraint changes result in jamming due to incompatible kinematics. This occurs as follows. Define the outward normal η_i of a curve γ_i to point into free space F. Let $\hat{t}$ be a unit vector in the positive t-direction. Jamming occurs at $\gamma_i \cap \gamma_j$ when both the inner products

$$\eta_i \cdot \hat{t} \quad \text{and} \quad \eta_j \cdot \hat{t} \tag{3.3}$$

are negative. At this point the simulation is terminated, because further motion is impossible.

Pure translational collision events can occur where a curve γ_i intersects the line $u = 0$. *Sliding collisions* can occur when two boundary curves of F intersect, i.e. at $\gamma_i \cap \gamma_j$. Jamming events can occur when both normals at $\gamma_i \cap \gamma_j$ point in the $(-t)$-direction. A non-jamming sliding collision causes a change of constraint (i.e. the sweep point now follows γ_j instead of γ_i). It is clear that sweep events of type (i), (ii) and (iii) are local geometric conditions and can be detected and handled while sweeping the line $L(t)$ over F. Similarly, it is clear that modifying the trajectory $z(t)$ at a sweep event can be done in $O(1)$ time.

We now describe the sweep events (iv) *snapping free from* and (v) *jamming on a single constraint*. Suppose the sweep point is following a constraint curve γ. A singularity occurs at vertical tangencies of γ. Assume, without loss of generality, that γ lies in the halfplane $u > 0$. There are two possibilities. If the F is concave at the singularity, then the sweep point has been following the "upper" branch of the curve. After the singularity, the sweep point follows the vector field attracting it towards $u = 0$. That is, the sweep point moves parallel to the u-axis toward the t-axis. It stops at the first new constraint curve it hits while moving away from the singularity towards the line $u = 0$. If no constraints are encountered, it stops at $u = 0$. This motion corresponds to the pawl "snapping free" from a single constraint edge. It executes an instantaneous pure rotation towards the zero position. If another constraint is in the way, then it stops there.

If F is convex at the singularity, then no further motion is possible, and the motion jams there on a single constraint. At this point the simulation is terminated. Singularity (vertical tangency) is a local geometric condition that can be detected during the plane sweep of F, since each curve is algebraic.

There is one more kinematic sweep event that is "dual" to type (iii) jamming due to incompatible kinematics. It is type (vi) *snapping free from a vertex*. It occurs at a constraint change $\gamma_i \cap \gamma_j$ (i.e. the sweep point is following a curve γ_i, and it hits another curve γ_j). However, in this case, both the outward normals η_i and η_j point in the positive t-direction. That is, the dot products in (3.3) are both positive. In this case, the sweep point "snaps free" from $\gamma_i \cap \gamma_j$ and moves vertically towards the attractor $u = 0$. The snapping free happens just as in event (iv) above. Snapping free from a vertex corresponds to the situation where suddenly there are no holonomic constraints on the pawl, so it can move towards its rest position $u = 0$. Conceptually, there is little difference from event (iv) (snapping free from one constraint). The sweep events of type (iv), (v) and (vi) are local geometric conditions and can be detected and handled while sweeping the line $L(t)$ over F.

3.2. Computing Friction Events

When the pawl is in contact with the environment, it is possible for the motion to stop because the contact forces are adequate to balance the applied forces. This is called "sticking due to friction". The algorithm must determine if this can happen during motion. The works of Mason (1982) and Erdmann (1984) address this issue in considerable detail.

In Donald and Pai (1991), we show that for our problem this determination can be expressed by simple algebraic constraints. The possible locations of the hinge point P_h relative to the contact point and edge can be divided into four sectors or qualitative dynamic regions. We can show that the question of whether or not a pawl will stick on the environment can be reduced to that of determining the sector containing the hinge point. In addition, if the hinge point is in a sliding sector, the state of the system will evolve until it enters the next sticking sector. The boundaries between the sliding and sticking sectors can be translated into configuration space constraints. For example, for type-B contact, the constraints are given by $R_\theta \boldsymbol{p}_i = \frac{|\boldsymbol{p}_i|}{1+\mu^2}(\pm \boldsymbol{n}_j + \mu \boldsymbol{v})$ where $\boldsymbol{v}$ is orthogonal to $\boldsymbol{n}_j$ and oriented in the direction of sliding.

Thus for each configuration space surface f_i we can define two constraints g_i and h_i which are also algebraic surfaces of the same degree as f_i; g_i and h_i depend on the direction of assembly $\dot{\boldsymbol{p}}$ in (2.1).

The surfaces g_i and h_i break up f_i into *sliding* and *sticking* regions. We call these *qualitative dynamical regions (QDRs)*. In a sliding region, motion is possible as t increases. In a sticking region, equilibrium results, and no further motion is possible (compare work on translational compliant motion, e.g. Donald, 1988; Briggs, 1989). Now, when we intersect f_i with the plane P_Y to obtain a curve γ_i we obtain a 1D slice of these qualitative dynamic regions (sliding and sticking).

Now, we define a seventh type of sweep event, (vii) a *sticking event* as follows. Suppose the sweep point is following a curve γ_i. If it enters a sticking region on the curve, then equilibrium is reached and the simulation is terminated. Entry into the sticking region corresponds to crossing another algebraic curve h_i or g_i, and hence is a local geometric event that can be detected and handled during the sweep. Note that g_i and h_i apply only to f_i and do not affect any other surface f_j, and hence we call them *local dynamic constraints.*

Finally we must slightly modify our kinematic plane sweep. After a pure translational or pure rotational collision with a curve γ_i, we first check to see whether we are in a sticking or sliding region on that curve. If it is a sticking region, we terminate the simulation in equilibrium, otherwise we proceed as above.

4. Application to Design for Assembly

In design for assembly, we often desire "locking" parts that, when mated, cannot be disassembled by motion plans in a particular family of directions. More generally, we may require interlocking parts that cannot be disassembled at all, for any translational motion plan. Most generally, one might want parts that cannot be disassembled without exerting large forces. Following a suggestion of Mason (1984), we call such objects "motion diodes". The term is motivated by the fact that motion is possible in certain directions, but not in others. Our usage differs from Mason's, in that his motion diodes are geometries from which a robot cannot be *guaranteed* to emerge. Our motion diodes are (flexible object, environment) pairs such that for some family of controls (or perhaps all controls) no change in the sign of the controls can reverse the motion to re-achieve the start position. In our simple case a "plan" is a translation given as a straight line motion $\boldsymbol{p} = \boldsymbol{p}_0 + \dot{\boldsymbol{p}}t$ where t is the time, $\boldsymbol{p}_0$ is the initial position (at $t = 0$), and $\dot{\boldsymbol{p}}$ is the root velocity. A family of controls corresponds to a set of velocities $\{\dot{\boldsymbol{p}}\}$, and changing the sign amounts to specifying $-\dot{\boldsymbol{p}}$.

If motion diodes can be designed, analyzed and verified, then they can be rigidly attached as "fasteners" to bodies that we wish to mate, but not to disassemble. Our algorithm can analyze designs for these kinds of diodes. For example, if the triangular pawl in figure 4 is attached in the z-axis (perpendicular to the figure) to a root body in a parallel x-y plane to the figure, then the root body can be fastened irreversibly to its mating part.

In the example depicted in figure 4 the triangular pawl can pivot about its center; a torsional spring is attached at the pivot. The root body is not in the plane of the pawl and is not shown. The pawl is moved down in a pure $-y$ translation, and in response to the reaction forces from the environment, it rotates compliantly. Let us label the black obstacles, starting with the uppermost one, in clockwise order, A, B and C. The pawl contacts A and rotates counterclockwise while sliding along A's upper left corner. Eventually, the pawl breaks contact with A, and snaps off, only to hit the rightmost vertex of C. It briefly slides (while rotating compliantly) along C, until it hits B. The tighter constraint from B takes over, and the pawl is again "cocked" counterclockwise until it breaks contact at the lower left vertex of B. Finally, the pawl snaps off B to its rest position.

Now, this mechanical system is a "diode" with respect to pure y-translation (see the figures). When the pawl is moved back up in the $+y$ direction, it jams due to incompatible kinematic constraints. More interestingly, if B and C are extended to the right and left (resp.), the system is a diode with respect to *all* translational motions. That is, no commanded translation can bring the flexible body back out of the hole between B and

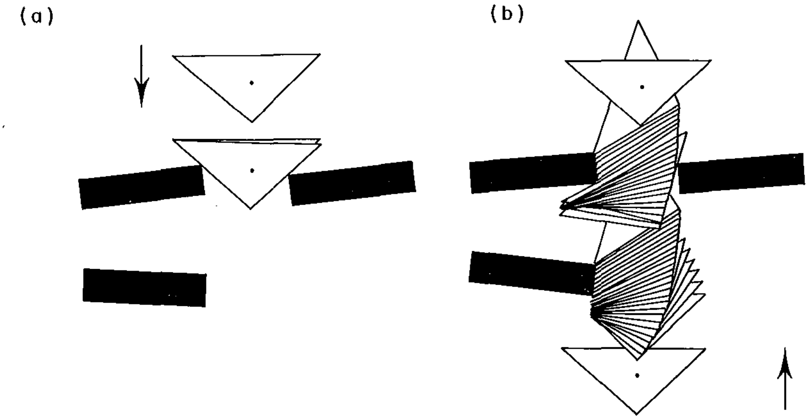

Figure 4. "Motion diode" example. There is no root body, and only one triangular pawl which can pivot about a torsional spring at its center. In figure (a), the pawl snaps off the upper right obstacle and continues downward. Figure (b) shows the reverse motion, during which the pawl gets stuck due to incompatible kinematic constraints.

C. Our algorithm can be used to decide that for a particular motion plan, a system is a diode.

5. Conclusion

There are several important situations in design where there is a need for rapid simulation of the motion of flexible objects using simplified models. Such simulations have typically required computationally intensive numerical simulation methods which are inappropriate for interactive design and for design search.

We have outlined an approach to fast simulation based on algebraic methods for a class of simple planar mechanical systems (Donald and Pai, 1991). In addition to speed, the algebraic approach provides precise combinatorial bounds on the complexity of the simulation and provides a systematic basis for enumerating special cases. While we can not claim that such an approach is suitable for all problems, we believe that this concept of "simulation as sweep" is a useful and broadly applicable paradigm.

References

A. Briggs (1989), "An efficient algorithm for one-step compliant motion planning with uncertainty", *Proc. 5th ACM Symp. Computational Geom.*, Saarbruchen, Germany, 187–196.

R. Brost (1989), "Computing metric and topological properties of configuration space obstacles", *Proc. IEEE Int. Conf. Robotics and Automation*, Scottsdale, AZ, 170–176.

J. Canny (1986), "Collision detection for moving polyhedra", *IEEE Trans. Patt. Analysis Mach. Intell.*, **8**(2), 200–209.

J. Canny (1989), "On computability of fine motion plans", *Proc. IEEE Int. Conf. Robotics and Automation*, Scottsdale, AZ, 177–182.

B.R. Donald (1987), "A search algorithm for motion planning with six degrees of freedom", *Artif. Intell.*, **31**(3), 295–353.

B.R. Donald (1988), "The complexity of planar compliant motion planning under uncertainty", *Proc. ACM Symp. Computational Geom.*, Urbana, IL, 309–318.

B.R. Donald (1990), "Planning multi-step error detection and recovery strategies", *Int. J. Robotics Research*, **9**(1), 3–60.

B.R. Donald and D.K. Pai (1991), "The motion of compliantly connected rigid bodies in contact", *Int. J. Robotics Research*, **10**, to appear.

M.A. Erdmann (1984), *On Motion Planning with Uncertainty*, Technical Report 810, AI Laboratory, MIT, Cambridge, MA.

M.A. Erdmann (1986), "Using backprojections for fine motion planning with uncertainty", *Int. J. Robotics Research*, **5**(1), 19–45.

J. Friedman, J. Hershberger and J. Snoeyink (1989), "Compliant motion in a simple polygon", *Proc. 5^{th} ACM Symp. Computational Geom.*, Saarbruchen, Germany, 175–186.

L. Guibas, M. Sharir and S. Sifrony (1988), "On the general motion planning problem with two degrees of freedom", *Proc. ACM Symp. Computational Geom.*, Urbana, IL, 289–298.

T. Lozano-Pérez (1983), "Spatial planning: A configuration space approach", *IEEE Trans. Comput.*, **C-32**(2), 108–120.

T. Lozano-Pérez, M. Mason and R. Taylor (1984), "Automatic synthesis of fine-motion strategies for robots", *Int. J. Robotics Research*, **3**(1), 3–24.

M.T. Mason (1982), *Manipulator Grasping and Pushing Operations*, PhD Thesis, MIT, Cambridge, MA.

M.T. Mason (1984), personal communication.

D.K. Pai (1988), *Singularity, Uncertainty and Compliance of Robot Manipulators*, PhD Thesis, Cornell University, Ithaca, NY.

D.E. Whitney (1982), "Quasi-static assembly of compliantly supported rigid parts", *J. Dynamic Syst. Measurement, and Control*, **104**, 65–77.