Submitted to 1990 IEEE Intl. Conf. on Robotics and Automation

(1)

On The Motion of Compliantly-Connected Rigid Bodies in Contact, Part I: The Motion Prediction Problem

Dinesh K. Pai *
Bruce R. Donald †

Department of Computer Science
Cornell University
Ithaca, NY 14853

Abstract

We consider the problem of planning and predicting the motion of a flexible object amidst obstacles in the plane. We model the flexible object as a rigid "root" body, attached to compliant members by torsional springs. The root's position may be controlled, but the compliant members move in response to forces from contact with the environment. Such a model encompasses several important and complicated mechanisms in mechanical design and automated assembly: snap-fasteners, latches, ratchet and pawl mechanisms, and escapements. The problem is to predict the motion of such a mechanism amidst fixed obstacles. For example, our algorithm could be used to determine whether a snap-fastener design can be assembled with a certain plan.

In this paper we analyze the physics of these flexible devices, and develop the first combinatorially precise algorithm for predicting their movement under a motion plan. Our algorithm determines when and where the motion will terminate, and also computes the time-history of contacts. In addition to providing

*Supported in part by DARPA grant N0014-88-K-0591, MSI grant U03-8300, NSF grant DMC-86-17355, and by ONR grant N00014-86-K-0281.

†This paper describes research done in the Computer Science Robotics Laboratory at Cornell University. Support for our robotics research is provided in part by the National Science Foundation under grant No. IRI-8802390 and by a Presidential Young Investigator award, and in part by the Mathematical Sciences Institute.

the first known exact algorithm that addresses flexibility in motion planning, we also note that our approach to compliance permits an exact algorithm for predicting motions under rotational compliance, which was not possible in earlier work.

1 Introduction

Previous work on algorithmic motion planning has largely concentrated on the movement problem for rigid bodies [LP83,B+82,Yap87]. Recently, work in compliant motion planning under uncertainty [LMT84,Erd84,Don88b] has focussed on the problem of moving rigid objects (e.g. pegs) in contact (i.e., compliantly) with obstacles (such as holes) under force-control, subject to bounded uncertainty and error. However, real objects that robots might assemble are typically not rigid. For example, a Sony Walkman is made of plastic parts that snap together. Significant advances were made in the design of the IBM ProPrinter, by replacing traditional fasteners such as screws with plastic parts that simply snap together. The reason these plastic parts snap together is that they are *flexible*: more precisely, they are *passively compliant*. This means that when the parts are brought together and an external force applied, the parts deform in a prescribed way. More interestingly, the force required to mate two parts may be much less than the force required to take them apart.

Since we wish to be able to design and have our robots assemble such objects given task-level descriptions, we must have a systematic program for reasoning about and predicting their motions in contact. To this end, we will first make precise a sufficiently powerful notion of flexibility to model the objects above. Our model is based on the well known *generalized spring* in the robotics and controls literature (*eg.*, [Mas81]). We will model the physics of interaction of the flexible parts and the obstacles using quasi-static analysis and Coulomb friction. Then, using these tools, we will proceed to develop a combinatorially precise algorithm for predicting the motion of a flexible object amidst obstacles. The algorithm is roughly $O(N^2 \log N)$ in the geometric complexity, N. A system based on the results of this paper is described in the companion paper [DP89], which also discusses its relevance to the design of such mechanisms.

From a complexity-theoretic standpoint, our result may also be viewed as follows. Earlier work on compliant motion planning [LMT84,Erd84,Don88b] has employed

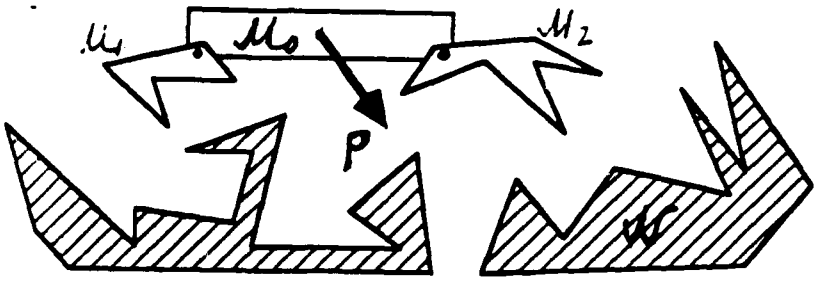

Figure 1: Linked body $\mathcal{M}$ moving among $\mathcal{N}$.

a dynamic model called the *generclized damper* [Whi76]. While this model has led to exact algorithms in the case of pure translations [Don88b,Can89], so far we have not been able to provide combinatorially precise algorithms once rotations are permitted. This is true even if the commanded motion is a pure translation and the moving object (eg., the peg) is merely rotationally compliant. One reason for this difficulty seems to be that under the damper model, the resulting motions are not obviously algebraic; even more disturbing, the outer envelope (or *forward projection*) of these motions cannot yet be algebraically described. (This is also a problem in many more complicated dynamical systems). However, under the model in this paper, rotationally compliant subparts are permitted, and yet all resultant trajectories of the system are algebraic. Finally, we believe that our algorithm is sufficiently simple that it could be implemented, and used to plan and verify the design of flexible mechanisms for ease of assembly.

1.1 Problem Statement

We consider the problem of moving a flexible, linked body $\mathcal{M}$ in the plane, in a polygonal environment $\mathcal{N}$. The flexible body $\mathcal{M}$ is constructed as follows: polygons $\mathcal{M}_h$, $h = 1, \ldots, l$, called "pawls", are attached 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 *Motion Prediction 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 of the Motion Prediction Problem are considered in our companion paper [DP89], including the determination of the time history of forces during the motion, 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. We shall assume that this translation is a straight line motion given by $\boldsymbol{p} = \boldsymbol{p}_o + \dot{\boldsymbol{p}}t$ where t is the time, $\boldsymbol{p}_o$ is the initial position (at $t = 0$) of P_h, and $\dot{\boldsymbol{p}}$ 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, [Whi82,Mas82,Don88a,Pai88]).

- 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 $\boldsymbol{p}$ 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. This assumption may appear to contradict the assumption of quasi-static motion, but is in fact practically a consequence. 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 $\boldsymbol{p}$ and the rotation (e.g., see [Can86]), 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.

Two types of contact are possible between the pawl $\mathcal{M}_h$ and a polygon in $\mathcal{N}$. Following the convention of Lozano-Pérez [LP83] and Donald [Don87], 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 [Can86]. 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, $\boldsymbol{n}_i$, and its distance to the hinge point along the normal, d_i. Let $\boldsymbol{p}_j$ 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

$$(\boldsymbol{p}_j - \boldsymbol{p}) \cdot R_\theta \boldsymbol{n}_i - d_i = 0. \tag{1}$$

Similarly, the type-B constraint can be written as

$$(R_\theta \boldsymbol{p}_i + \boldsymbol{p}) \cdot \boldsymbol{n}_j - d_j = 0. \tag{2}$$

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

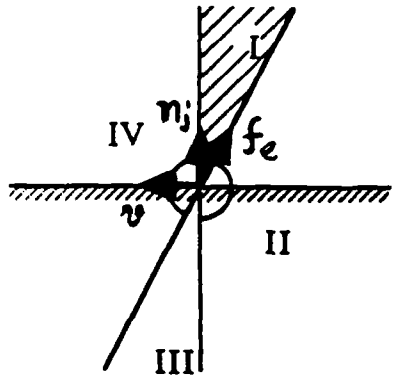

Figure 2: Geometric Interpretation of Coulomb's Law

2 Sticking due to Friction

A careful analysis of the physics of compliant sliding is necessary to formulate a precise, combinatorial version of this apparently continuous problem. 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," to distinguish it from other kinds of sticking (see Section 5). The algorithm must determine if this can happen during motion. The works of Mason [Mas82] and Erdmann [Erd84] address this issue in considerable detail. In this section, we show that for our problem this determination can be made by a simple geometric test.

Coulomb's Law of friction states that the tangential force due to friction, f_t, on a point which is sliding on an edge in a specified direction is given by $f_t = \mu f_n$, where f_n is the force normal to the edge and μ is the coefficient of friction. In addition, the direction of the friction force is opposite that of the motion.

Coulomb's Law has a geometric interpretation which is frequently more useful (see, for example, [Mas82]). We provide here a slightly different interpretation, which is more applicable to our problem. Suppose point p_i is in contact with edge e_j. The outward pointing unit normal vector to the edge is denoted n_j. We are also given the direction of sliding by specifying v, the unit vector tangent to the edge in the direction of sliding (see Figure 2). Coulomb's Law defines a half-cone of forces acting on the point p_i. The cone is the convex combination [1] of the vectors

[1]The convex combination of two vectors x and y is the set of all vectors of the form $\zeta x + \eta y$, where $\zeta, \eta \geq 0$.

n_j and the vector $f_e = n_j - \mu v$. Any force in the interior of this cone and along n_j can be resisted and the point will stick. A force along f_e will result in p_i sliding in equilibrium, in the direction of v. A force outside the cone results in the point sliding along v, out of equilibrium, which is precluded by our assumption of quasi-static motion.

We first consider the type-B contact: the point is the vertex i of the pawl, and the edge is the edge e_j of the environment. Let $r = R_\theta p_i$. The assumption that stable contact is maintained (see Section 1.1) between the point and the edge restricts the possible values of the force f on the pawl and the torque $r \times f$ [2] due to the contact force. First, $f \cdot n_j$ must be positive; second, if the hinge is in the right half-plane, we require $r \times f < 0$ for stability, and if it is in the left half-plane, we require that $r \times f > 0$. For quasi-static sliding, the total force is given by $f = \alpha f_e$, $\alpha > 0$. Therefore f trivially satisfies the condition $f \cdot n_j > 0$. The condition on the torque $r \times f$, however, depends on the location of the hinge point. Let the possible locations of the hinge point with respect to the contact point, $-r$, be divided into four sectors as follows: if $-r = \beta n_j + \gamma f_e$, the hinge point is in Sector

I if $\beta > 0, \gamma > 0$,
II if $\beta < 0, \gamma > 0$,
III if $\beta < 0, \gamma < 0$,
IV if $\beta > 0, \gamma < 0$.

Figure 2 also indicates the sectors.

Proposition 1. For type-B contact, the contact point will slide when the hinge is in sectors II and IV. The contact point will not slide (will stick due to friction) when the hinge is in sectors I and III.

Proof. Suppose the pawl is sliding quasi-statically. Then,

$$\begin{aligned} f &= \alpha f_e \\ &= \alpha(n_j - \mu v). \\ r &= -[\beta n_j + \gamma f_e] \\ &= -[(\beta + \gamma) n_j - \gamma \mu v]. \end{aligned}$$

[2] Here $r \times f$ is the component of the usual cross product along the axis orthogonal to the plane, i.e., in coordinates, $r \times f = r_x f_y - r_y f_x$.

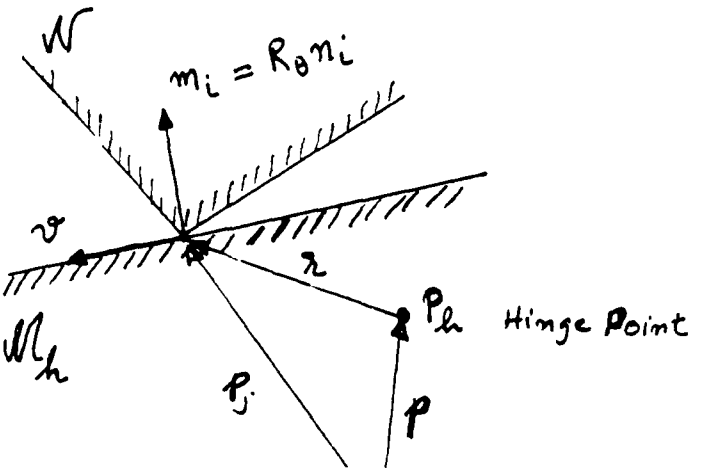

Figure 3: Type-A Contact

$$\begin{aligned} r \times f &= -\alpha[0 - (\beta+\gamma)\mu + \gamma\mu + 0] \\ &= \alpha\beta\mu. \end{aligned} \tag{3}$$

Since $\alpha, \mu > 0$, the sign of $r \times f$ is just that of β. Now, in sectors I and II, the right-half plane, we require that $r \times f < 0$ to maintain stable contact. But from Equation 3, $r \times f > 0$ in sector I and $r \times f < 0$ in sector II. Hence the contact point will stick in sector I and slide in sector II. A similar argument shows that the contact point will stick in III and slide in IV. □

A similar argument applies to type-A contact if the sectors and related terms are defined appropriately (see Figure 3). Let f be, as before, the force on the point of contact, which is now a vertex j of the environment, and let r be the vector from the hinge point to the contact point, which is now given by $r = p_j - p$. Let the outer normal of the edge e_i be denoted by $m_i = R_\theta n_i$, v be the unit vector tangent to the edge in the direction in which the vertex slides with respect to the pawl's edge and $f_e = m_i - \mu v$. To maintain contact, $f \cdot m_i$ must be positive and, if the hinge is in the right half-plane, we require $r \times -f > 0$, i.e. $r \times f < 0$, and if it is in the left half-plane, we require that $r \times f > 0$.

With the terms defined as above, we can divide the possible locations of the hinge point with respect to the contact geometry into four sectors exactly as we did for Type-B contact, despite the fact that the interpretation of the sectors is different. Now it is easy to prove the corresponding proposition:

Proposition 2. For type-A contact, the contact point will slide when the hinge is in sectors II and IV. The contact point will not slide (will stick due to friction) when the hinge is in sectors I and III.

Proof. The proof is textually identical to the proof for Type-B contact, with n_j replaced by m_i. □

3 The Sliding Direction

The analysis of Section 2 utilized the direction of motion to determine whether the pawl could stick due to friction. It turns out that the direction of the motion can be determined from the differential kinematics of the contact, given the motion of the hinge, p.

Consider the motion for a type-B contact. For convenience, assume that the Y-axis of the reference coordinate frame is along the outward normal of the edge, i.e., $y = n_j$, and the X-axis is along the edge, oriented to form a right-handed coordinate system. Recall that $r = R_\theta p_i$. Then, differentiating the type-B kinematic constraint equation, Equation 2, we get

$$(\dot{r} + \dot{p})\cdot n_j = 0. \tag{4}$$

Now $\dot{r} = \omega(-r_y, r_x)^T$, where ω is the angular velocity of the pawl. Hence, the above equation can be written as

$$\omega r_x + \dot{p}_y = 0. \tag{5}$$

This equation can be derived in another way, which will be useful in Section 5. Define the *departure speed*, u, as the component of the velocity of the contact point along the edge normal n_j. The velocity of the contact point is given by $\dot{r} + \dot{p}$. Hence

$$u = (\dot{r} + \dot{p})\cdot n_j = \omega r_x + \dot{p}_y. \tag{6}$$

The differential kinematic constraint for sliding corresponds to the requirement that the departure speed is zero. It is easy to see that the two derivations are equivalent.

The *sliding speed*, s, is defined as the component of the velocity of the contact point in the x direction (the vector v of Section 2 is therefore $sgn(s)x$). Hence,

$$s = (\dot{r} + \dot{p})\cdot x \tag{7}$$

$$= -\omega r_y + \dot{p}_x. \tag{8}$$

Combining with Equation 5,

$$s = \dot{p}_x + \frac{\dot{p}_y}{r_x} r_y. \tag{9}$$

Equation 9 has a simple geometric interpretation, shown in Figure 4. The circle

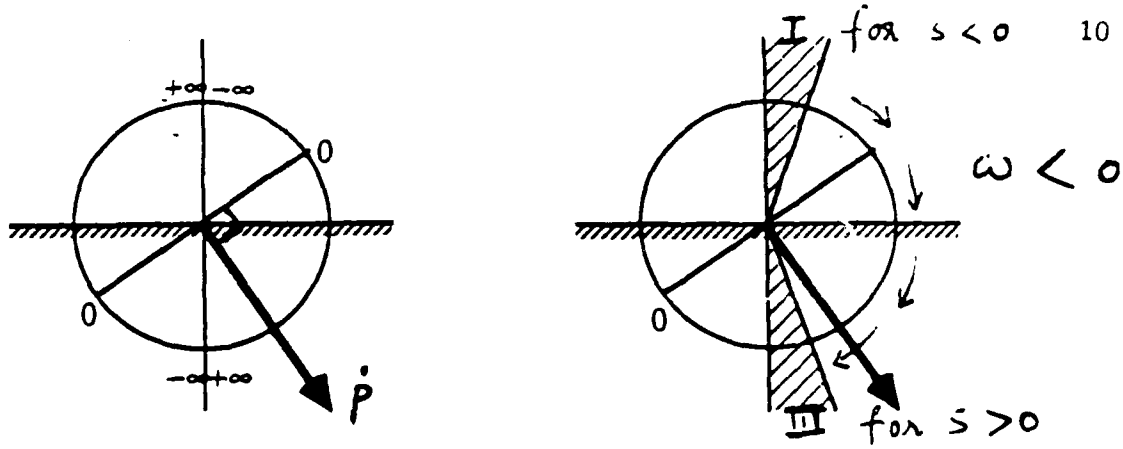

Figure 4: The sliding direction for Type-B contact.

represents the possible positions of the hinge point with respect to the contact point. The numbers on the circle indicate the sliding speed, when $\dot{p}_y < 0$ (if $\dot{p}_y > 0$, the speeds shown have opposite sign; if $\dot{p}_y = 0$, s is constant). Notice that the sliding speed is 0 when $\boldsymbol{r} \cdot \dot{\boldsymbol{p}} = 0$. Also, from Equation 5, $\omega < 0$ when the hinge point ($-\boldsymbol{r}$) is in the right half-plane of the figure, and $\omega > 0$ in the left half-plane.

We can now combine the above results with those of section 2. Suppose, without loss of generality, that the hinge point is in the right half-plane. If the hinge is in the region corresponding to $s < 0$, the possible direction of slide is to the left. If, in addition, the hinge is in sector I, it will stick. If it is outside sector I, the location of the hinge will move in the direction of ω, at some point (corresponding to $\boldsymbol{r} \cdot \dot{\boldsymbol{p}} = 0$), the direction of slide will reverse to $s > 0$. This also brings sectors III and IV to the right half-plane. Eventually, the hinge will enter sector III for $s > 0$, and stick. Figure 4 also depicts the evolution of the hinge point.

The time at which the contact will stick in sector III is easily computed by letting $R_\theta \boldsymbol{p}_i = \frac{|\boldsymbol{p}_i|}{1+\mu^2}(\boldsymbol{n}_j + \mu \boldsymbol{v})$ in the constraint equation 2 and solving the resulting linear equation for t.

A similar but slightly more complicated analysis applies to Type-A contact. We shall first derive an expression for the velocity $\boldsymbol{w}$ of the contact point with respect to a frame fixed on the moving pawl, and expressed in the world coordinate frame. We shall assume for simplicity of derivation, that the coordinate frame attached to the pawl has its Y-axis along the edge normal $\boldsymbol{m}_i = R_\theta \boldsymbol{n}_i$, and the X-axis is along the edge so as to form a right-handed coordinate system. As in Section 2, for Type-A contact, $\boldsymbol{r} = \boldsymbol{p}_j - \boldsymbol{p}$, and let ${}^p\boldsymbol{r} = R_\theta{}^T \boldsymbol{r}$ be the corresponding vector in the

coordinate frame fixed to the pawl. Let Ω be the skew-symmetric angular velocity matrix, given by

$$\Omega = \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix}. \tag{10}$$

where ω is the angular velocity of the pawl. From elementary kinematics (see, for example, [BR79]), $\Omega = \dot{R}_\theta R_\theta{}^T$.

The velocity in the frame attached to the pawl, $R_\theta{}^T w$, can now be calculated by differentiation:

$$\begin{aligned} R_\theta{}^T w &= {}^p\dot{r} \\ &= R_\theta{}^T \dot{r} + \dot{R}_\theta{}^T r \\ &= R_\theta{}^T \dot{r} + R_\theta{}^T \Omega^T r. \\ &\quad \text{Therefore,} \qquad (11) \\ w &= \dot{r} + \Omega^T r. \qquad (12) \end{aligned}$$

The differential constraint corresponds to the departure speed, u, being zero. Therefore,

$$\begin{aligned} u &= w \cdot m_i \\ &= \dot{r}_y - \omega r_x \\ &= 0. \qquad (13) \end{aligned}$$

Solving for ω,

$$\begin{aligned} \omega &= \frac{\dot{r}_y}{r_x} \\ &= \frac{-\dot{p}_y}{r_x}. \qquad (14) \end{aligned}$$

Hence, the sliding speed, s, is given by

$$\begin{aligned} s &= w \cdot x \\ &= \dot{r}_x + \omega r_y \\ &= \dot{r}_x + \frac{-\dot{p}_y}{r_x} r_y \\ &= -\dot{p}_x - \dot{p}_y \frac{r_y}{r_x} \qquad (15) \end{aligned}$$

The map of sliding speeds for Type-A contact is almost identical to Figure 4, except that the direction of the vector $\dot{p}$ is reversed. However, the orbit of the hinge

point during sliding is more complicated than in the type-B case. The time at which the hinge point will enter Sector III and stick can be computed as follows. At the time of entry into Sector III, the r is given by

$$\begin{aligned} r &= \frac{|r|}{\sqrt{1+\mu^2}}(m_i - \mu v) \\ &= \frac{|r|}{\sqrt{1+\mu^2}}\left(m_i + \mu \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} m_i\right) \\ &= \frac{|r|}{\sqrt{1+\mu^2}}\begin{pmatrix} 1 & -\mu \\ \mu & 1 \end{pmatrix} m_i \end{aligned} \tag{16}$$

Hence,

$$m_i = \frac{1}{|r|\sqrt{1+\mu^2}}\begin{pmatrix} 1 & \mu \\ -\mu & 1 \end{pmatrix} r \tag{17}$$

Plugging this back in Equation 1, we get

$$\begin{aligned} r \cdot m_i - d_i &= \frac{1}{|r|\sqrt{1+\mu^2}} r^T \begin{pmatrix} 1 & \mu \\ -\mu & 1 \end{pmatrix} r - d_i \\ &= \frac{|r|}{\sqrt{1+\mu^2}} - d_i = 0. \end{aligned} \tag{18}$$

(19)

Since r is a linear function of t, this is a quadratic equation in time and can be solved for t.

4 Intersection Problems and Collision Detection

As the linked body $\mathcal{M}$ is moved in the environment $\mathcal{N}$, the pawl $\mathcal{M}_h$ may collide with an obstacle. We need robust algorithms to detect and report the collision. While a pawl is sliding on a feature of the environment, it may arrive at a configuration at which it is kinematically impossible to continue to maintain contact – the pawl will either snap off the environment or jam on it. We need a way to determine when this occurs. These two seemingly unrelated problems are in fact reducible to intersection problems in configuration space.

The geometric intersection algorithms for our problem, however, involve several subtle and important issues to make them robust in practice. These issues have a somewhat different flavor than the rest of this paper. Hence we discuss them in detail in the companion paper [DP89], which also examines a host of important

practical considerations in the implementation of geometric intersection algorithms. We shall briefly sketch the methods used, for the sake of completeness.

Three types of motion are possible for the pawl in our problem: pure translation without rotation; pure rotation without translation; and most importantly, the motion of the pawl sliding while maintaining contact with a given feature of the obstacle. The collision detection problem is to find the times at which the pawl collides with $\mathcal{N}$, and report which features are in contact during collision.

In each of the cases, the obstacle constraint equations (1 and 2) and can be reduced to algebraic equations. This is possible since the trigonometric functions in the rotation matrix R_θ can be replaced by new indeterminate quantities (eg. $\sin\theta = s$, $\cos\theta = c$), with the addition of a new constraint ($s^2 + c^2 - 1 = 0$), as in [Don87]. More practically, the substitution $u = \tan\frac{\theta}{2}$ 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. If m is the geometric complexity of the pawl and n is that of the size of the environment, there are mn such constraints, and the total time complexity for collision detection is $O(mn)$.

Since the constraint equations are quadratic in u, it possible that in some interval of time the discriminant Δ of the constraint equation (treated as an quadratic in u) becomes negative, yielding no solution for u. This corresponds to the pawl either having snapped off the constraint or having jammed on it. By solving for the critical times t_s at which the discriminant changes sign, we can determine, by careful analysis, which case occurs. The details of this analysis are found in [DP89], since it is related to the problem of choosing the "correct branch" for collision detection.

5 Resolving Multiple Contacts

When the pawl is sliding on the environment, it is possible for a second contact to occur due to collision. Generically, the pawl can maintain two contacts for only an instant in time. The algorithm has to determine if subsequent motion is possible,

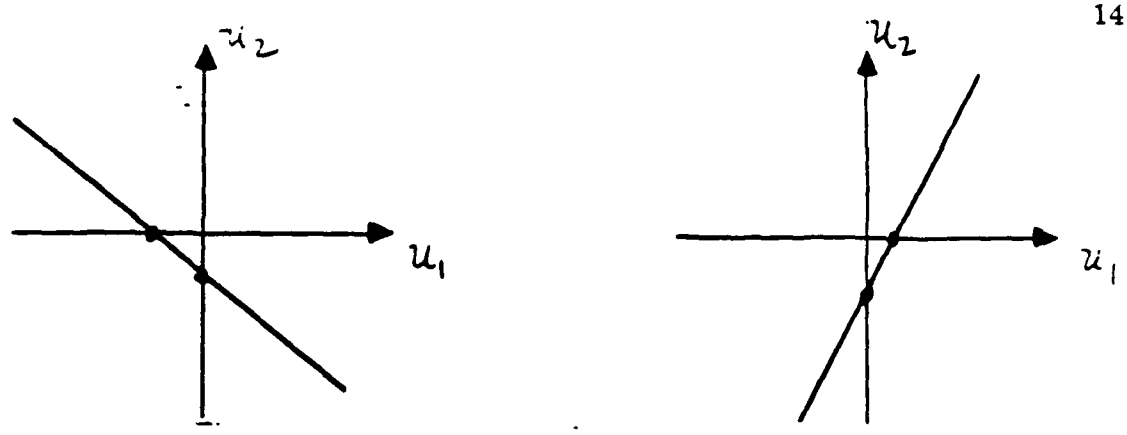

Figure 5: Possible lines in the u_1–u_2 plane.

and if so, which of the two contacts remain. In this section we show that the second contact either causes the pawl to stick due to incompatible constraints, or the second contact replaces the first. We also provide the means for determining which of the two cases occurs.

We shall first consider the case when both contacts are of type-B. As in section 3, we define the *departure speed*, u, by

$$u = (\dot{\mathbf{r}} + \dot{\mathbf{p}})\cdot\mathbf{n}_j = \omega r_x + \dot{p}_y. \tag{20}$$

Let us subscript the quantities corresponding to the original contact by 1, and the the quantities for the new contact by 2. Since both the contacts are on with the same pawl, they have the same angular velocity ω. Hence we have:

$$\omega = \frac{u_1 - \dot{p}_{y1}}{r_{x1}} = \frac{u_2 - \dot{p}_{y2}}{r_{x2}} \tag{21}$$

This equation specifies a linear relationship between u_1 and u_2 which must hold for all values of angular velocity. However, not all possible values of the coefficients can occur. If the new contact occured while the contact point was sliding on the interior of an edge, $u_1 = 0$ and the departure velocity u_2 must be negative (recall that $\mathbf{n}_j$ is the *outward* normal). Hence only a line for which the intercept with the u_2 axis is negative is possible (see Figure 5). Of these, if the u_1-intercept is negative, it is clear that it is not possible to find an angular velocity such that neither of the departure velocities are negative. Hence the pawl must stick due to incompatible constraints. Figure 6 depicts a situation in which such sticking occurs. Similarly, if the u_1-intercept is positive, the only possible way to maintain contact with the

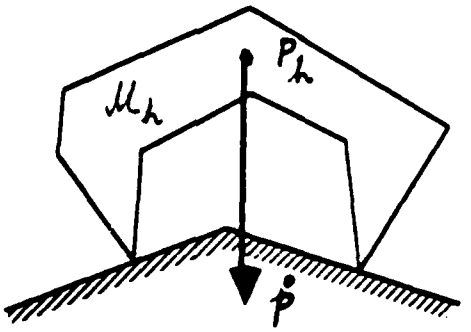

Figure 6: Sticking due to incompatible constraints.

environment is to have $u_2 = 0$ (sliding along the second contact), while $u_1 > 0$ (the first contact departing from its edge).

Finally, to ascertain which of the two cases occur, we only need to check the sign of the u_1-intercept of the line. From equation 21, this is given by

$$sgn\left(\dot{p_{y1}} - \dot{p_{y2}}\frac{r_{x1}}{r_{x2}}\right). \tag{22}$$

The other three possible transitions are similar. As before, the fact that departure speeds are linked by being on the same pawl leads to a linear relationship between u_1 and u_2, and as before, the pawl will stick due to incompatible kinematic contraints or the second contact will replace the first. We can check which of the two occurs by checking the sign of the u_1 intercept – if it is negative, the pawl sticks. We list the expressions for the sign of the u_1 intercept for the three possible cases below. When the two contacts are of different type, we shall superscript r by the contact type for clarity, since the interpretation of r depends on the contact type.

If both contacts are Type-A:

$$sgn\left(\dot{r^A_{y1}} - \dot{r^A_{y2}}\frac{r^A_{x1}}{r^A_{x2}}\right). \tag{23}$$

If the first contact is Type-B and the second contact is Type-A:

$$sgn\left(\dot{p_{y1}} + \dot{r^A_{y2}}\frac{r^B_{x1}}{r^A_{x2}}\right). \tag{24}$$

If the first contact is Type-A and the second contact is Type-B:

$$sgn\left(\dot{r^A_{y1}} + \dot{p_{y2}}\frac{r^A_{x1}}{r^B_{x2}}\right). \tag{25}$$

6 The Algorithm

Putting all the pieces developed in the previous sections together, we present the following algorithm for the motion of a single pawl. In order to keep the number of special cases small, we describe the algorithm for type-B contacts only. It is straightforward to handle type-A contacts as well.

Preprocessing: Determine the possible collisions of the pawl with $\mathcal{N}$ while undergoing a pure translation of $\boldsymbol{p}$. Let the first such time be t_b. Also determine the time t_e at which the root polygon $\mathcal{M}_0$ collides with $\mathcal{N}$. The algorithm can now be concerned only with the time interval $[t_b, t_e]$.

Step 1: For the current contact, determine the possible direction of slide, using the results of Section 3.

Step 2: Determine if sticking occurs due to friction, using the results of Section 2.

If it sticks, report this, and END.

Else, determine the time t_s at which the contact enters the sticking region, using the results of Section 3.

Step 3: Determine the time at which the pawl jams on the constraint due to the kinematics or snaps off, using the results of [DP89].

If it jams, report this, and END.

Else, the pawl has snapped off the constraint. The pawl now undergoes a pure rotation towards its rest position. Determine the next collision during this rotation.

- If a new contact is found, report it. GOTO Step 1 and repeat with the new contact.
- Else find the next collision under pure translation after the rotation has finished. Report the collision_found. GOTO Step 1 and repeat with the new contact. If no collision is found, END.

Step 4: Determine if other collisions between the pawl and $\mathcal{N}$ occur while the pawl is sliding while maintaining the current contact, using the results of Section 4. Let t_c be the time of the first such collision. Note that one also obtains the time at which the contact reaches the end of the current edge e_j, i.e., the time at which a collision occurs between the contact vertex and the adjacent edge. Let this time be t_f.

If $\min\{t_c, t_s\} = t_s$ then report that the pawl has stuck due to friction, END;

Else, if $t_c = t_f$, the contact vertex has reached the end of the current edge. The pawl now snaps back toward its rest position. Find the next collision as in Step 3. Note that the transition to an adjacent edge is included in this test.

Else, if $\min\{t_c, t_s\} = t_c$, report the new contact that has occurred. Analyze the multiple contacts using the results of Section 5.

- If the new contact is incompatible, report this, and END; the pawl has stuck due to incompatible kinematic constraints.
- Else, the new contact is effective. GOTO Step 1 and repeat with the new contact.

End

Each collision detection step above takes time $O(mn)$ [Don87]. In addition, suppose the contact changes c times. Then the loop of Steps 1–4 is repeated c times. Therefore the algorithm takes time $O(cmn)$. We expect (but do not prove) that $c = O(mn)$, and that it will typically be much smaller. However, in order to obtain an upper bound, we offer the following modification to the above algorithm, yielding a running time of $O(m^2n^2 \log mn)$.

We handle the collision detection while sliding via the following data structure: with each constraint equation, we associate two balanced search trees, labeled **Plus** and **Minus**. The **Plus** tree corresponds to contact with the constraint with positive *Branch-Sign*, while the **Minus** tree corresponds to contact with negative *Branch-Sign*. See [DP89]. Essentially, the *Branch-Sign* is used to choose the correct root in the solving the intersection problem for collision detection. For type-B constraints, for example, the *Branch-Sign* is the sign of $r \times n_j$. The *Branch-Sign* is invariant during the motion along the constraint. The leaves of each tree contain, sorted by the time t, all the tuples (t, k) where $k = 1, \ldots, mn$, is the index of a contact

constraint equation, and t is the time at which the constraint is satisfied by $\boldsymbol{p}(t)$. Generically, the number of different times t for a given k has a fixed upper bound which can be obtained from Bezout's Theorem [vdW50]. Hence, there are $O(mn)$ nodes in each tree, and hence each tree takes $O(mn \log mn)$ to build, and hence the entire data structure takes $O(m^2n^2 \log mn)$ time to build. Each collision detection step of the algorithm now takes $O(\log mn)$.

We claim that the collision detection can be requested at most as many times as there are leaves in the data structure. Suppose we request the next collision at time t, while maintaining constraint k_1 with a given *Branch-Sign*, and the answer is the tuple (t_c, k_2). From the algorithm it follows that only four things can happen:

1. The pawl sticks due to friction, or due to kinematic constraints, in the time interval $[t, t_c]$, and the program terminates.

2. The pawl snaps off the end of an edge at time $t_c = t_f$. This implies that leaf of the tree with time t_f and constraint corresponding to the contact vertex touching an adjacent edge are used up, and the time is advanced to t_f.

3. The pawl snaps off the interior of edge, since the contact can no longer be maintained with the constraint k_1. Hence both the subtrees under k_1 are used up, since the pawl can never return to k_1.

4. (t_c, k_2) is the new contact. Hence this leaf is used up and the time is advanced to t_c.

Hence, no tuple can be the answer to a collision detection request more than once, proving our claim. Since the entire data structure has $O(m^2n^2)$ leaves, the collision detection while sliding takes $O(m^2n^2 \log mn)$ time. We can similarly show that the collision detection problem for pure translation and pure rotation can be done in $O(m^2n^2 \log mn)$.

7 Conclusions

In this paper, we addressed the problem of predicting the motion of a flexible object amidst a polygonal environment. To this end, we presented a careful analysis of

the physics of the interaction of the flexible body with the environment. The analysis yielded several simple tests to determine the motion of the body, which were integrated into an algorithm for predicting the motion under a given motion plan.

This motion prediction problem is the first step in reasoning about such devices. The companion paper [DP89] describes the implementation of the algorithm and related issues, as well the significance of the algorithm for designing compliant parts for ease of assembly and the effect of uncertainty. Future work includes the incorporation of more complicated models of the flexible objects (such as that of [TM87]) and the dynamics of interaction.

Acknowledgments:: We would like to thank Randy Brost for very helpful comments on these ideas and on a draft of this paper, and to thank Mike Caine for helpful discussions.

References

[B+82] Brady et al. *Robot Motion: Planning and Control.* The MIT Press, 1982.

[BR79] O. Bottema and B. Roth. *Theoretical Kinematics.* North Holland, Amsterdam, 1979.

[Can86] John Canny. Collision detection for moving polyhedra. *IEEE Trans. PAMI*, 8(2), March 1986.

[Can89] John Canny. On computability of fine motion plans. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 177–182, 1989.

[Don87] B. R. Donald. A search algorithm for motion planning with six degrees of freedom. *Artificial Intelligence*, 31(3), 1987.

[Don88a] B. R. Donald. A geometric approach to error detection and recovery for robot motion planning with uncertainty. *Artificial Intelligence*, 37((1-3)):223–271, Dec. 1988.

[Don88b] Bruce R. Donald. The complexity of planar compliant motion planning under uncertainty. In *Proc. ACM Symposium on Computational Geometry*, June 1988.

[DP89] B. R. Donald and D. K. Pai. The motion of compliantly-connected rigid bodies in contact, part ii: A system for analyzing designs for assembly. Submitted for 1990 IEEE International Conference on Robotics and Automation, 1989.

[Erd84] Michael A. Erdmann. On motion planning with uncertainty. Technical Report 810, MIT AI Laboratory, 1984.

[LMT84] T. Lozano-Pérez, M. Mason, and R. Taylor. Automatic synthesis of fine-motion strategies for robots. *International Journal of Robotics Research*, 3(1), 1984.

[LP83] Tomás Lozano-Pérez. Spatial planning: A configuration space approach. *IEEE Transactions on Computers*, C-32(2):108–120, February 1983.

[Mas81] Matthew T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11(6), 1981.

[Mas82] Matthew T. Mason. *Manipulator Grasping and Pushing Operations.* PhD thesis, M. I. T., June 1982.

[Pai88] Dinesh K. Pai. *Singularity, Uncertainty and Compliance of Robot Manipulators.* PhD thesis, Cornell University, Ithaca, NY, May 1988.

[TM87] G. G. Trantina and M. D. Minnichelli. The effect of nonlinear material behaviour on snap-fit design. In *ANTEC '87*, pages 438–441, 1987.

[vdW50] B. L. van der Waerden. *Modern Algebra*, volume II. Frederick Ungar Publishing Co., 1950.

[Whi76] Daniel E. Whitney. Force feedback control of manipulator fine motions. In *Joint Automatic Control Conference*, pages 687–693, West Lafayette, Indiana, 1976.

[Whi82] Daniel E. Whitney. Quasi-static assembly of compliantly supported rigid parts. *J. Dynamic Systems, Measurement, and Control*, 104:65–77, March 1982.

[Yap87] C. Yap. Algorithmic motion planning. in *Advances in Robotics: Volume 1*, edited by J. Schwartz and C. Yap, Lawrence Erlbaum Associates, 1987.