Computational Complexity of Motion and Stability of Polygons

Richard S. Palmer
Ph.D Thesis

TR 89-1004
May 1989

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

Computational Complexity

of

Motion and Stability of Polygons.

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Richard S. Palmer

June, 1986

Computational complexity

of

motion and stability of polygons

Richard S. Palmer, Ph.D.

Cornell University 1986

The ability to model physical objects and procedures accurately enough to predict their behavior without performing physical experimentation is a fundamental goal of robotics. This facility is prerequisite to offline modeling of assembly tasks, high level robotics languages, and automated assembly planning.

This thesis defines and gives algorithms for two classes of physical modeling problems: *Mobility* problems and *Stability* problems.

The Mobility problem for polygons is that of determining whether, in a configuration of non-intersecting polygons, one or more polygons can be moved (relative to some other polygon in the configuration) without causing intersection. Mobility is shown to be NP-hard. An upper bound for the mobility problem remains an open problem.

Translational mobility is the problem of determining whether any polygons can be simultaneously moved by translations without causing intersection. Translational mobility is shown to be NP-complete.

Infinitesimal mobility is the problem of determining whether there is a set of velocities for the polygons of the configuration such that no point of a polygon P_i that is in contact with another polygon P_j has a velocity directed towards the interior of P_j. Infinitesimal mobility can be viewed as an approximation to the mobility problem

in that any configuration that is mobile is also infinitesimally mobile. Infinitesimal mobility is shown to be NP-complete.

The *Stability* problem for polygons with mass is the problem of determining whether a configuration of polygons is at a static equilibrium point. The stability problem is considered for configurations of polygons with and without friction, and is shown to be NP-hard for both cases. An algorithm is given that distinguishes between configurations that are stable, unstable, and indeterminate. The ability to distinguish indeterminate configurations is important because indeterminate configurations arise when the model of an assembly is not accurate enough to determine whether the assembly is stable.

Finally, a restricted class of configurations is developed, the *determined* configurations, for which a conservative stability problem can be solved in polynomial time. The determined configurations are a natural class in the sense that they preclude a type of contact that "seems unpredictable." If undetermined points are desired or unavoidable, the restricted stability problem can be solved in time exponential in the number of undetermined points in the configuration.

Biographical Sketch

Rick Palmer was born in Portland, Maine, on December 18, 1953, and lived in Lexington, Massachusetts, during most of his childhood. He attended Ohio University, Middlesex Community College, and the University of Massachusetts at Boston. He received the Bachelor of Arts degree in Computer Science from the University of Massachusetts at Boston in 1981.

In addition to Computer Science, he is interested in music, mechanical toys (especially fast cars), and viticulture.

Acknowledgements

I would like to thank Al Demers, my thesis advisor, for his encouragement, insight, and most of all for his enthusiasm. This thesis would certainly not have been possible without the many hours we spent discussing these problems.

I would also like to thank John Hopcroft for his support and knowledge of "what's interesting."

Dexter Kozen helped me by discussing problems related to linear and quadratic programming, which lead to several of the proofs in this thesis.

At the University of Massachusetts at Boston, Jim Perchik, Bob Morris, and Joan Lucas encouraged my interest in computer science. Tony Marcinkowitz showed me how to write good code.

Aliza Wechsler, B. Natarajan, Dean Krafft, and Christoph Hoffman criticized drafts of this thesis. Their comments, especially Aliza's and Nat's, were invaluable.

Gordon Wilfong suggested that a result like Theorem 4.13 would be interesting.

I would also like to thank my other friends, Gabi Bracha, Rogerio Drummond, Amr El Abbadi, Jim Hook, Kate MacGregor, Jim Mildrew, Gil Neiger, Turhan Rahman, T. K. Srikanth, and Solveig Torgersen for making my years at Cornell so enjoyable.

Contents

List of Figures

Chapter 1

Introduction

Robotics must move toward integrating the process of designing and building physical objects. Current robotic assembly planning makes little use of physical object models, is time consuming, and is subject to error.

This thesis investigates a fundamental modeling problem in robotics: determining whether a configuration of physical objects is *stable*, that is, guaranteed to remain immobile in a gravitational field.

In contrast to the extensive practical work done in computer aided design (CAD), where mechanical assemblies are routinely designed with computerized drafting systems using object models to test physical properties, robotic assembly is primitive and ad hoc. Most robotic assembly procedures are programmed by manually guiding a robot arm to the positions it will take in the assembly process and then writing control code in a low level language similar to BASIC.

Because objects and processes are not explicitly represented in such a language, it is difficult verify that an assembly plan is correct. A higher level robotics language would use object models to provide offline verification of each step in an assembly plan. Testing stability is essential to offline verification because each step in the plan must result in a stable configuration.

Ultimately, a robotics language could automatically generate many of the intermediate steps in an assembly procedure from a high level description, such as an exploded diagram. Besides testing stability, such a system would *propose* configurations. Since there are a infinite number of configurations for a given set of objects, the system would have to make intelligent decisions when proposing intermediate steps.

1.1 Summary of thesis

Chapter 1 introduces the stability problem. It discusses the fundamental issues in computer modeling of physical objects and processes, previous work in this area, and several problems that must be addressed before it will be possible to construct a general facility for offline modeling of robotic assembly procedures. Chapter 1 also describes a more difficult related problem: *planning* the construction of a given configuration.

Finally,Chapter 1 argues that a rigid model of physical objects is more appropriate than a deformable model for stability problems in robotics.

Chapter 2 contains the definitions of configurations and motion of polygons in the plane used in Chapter 3 and Chapter 4.

Chapter 3 shows that it is NP-hard to determine whether there is any feasible, nonintersecting, simultaneous (non-trivial) motion of a set of polygons in a given configuration in the plane. It also describes restricted mobility problems(translational mobility and infinitesimal mobility) and shows that these are NP-complete. Finally, it shows that mobility remains NP-hard even if all the polygons are convex.

Using the results from Chapter 3, Chapter 4 defines several variations of the stability problem for rigid polygons in the plane and shows that it is CO-NP-hard to determine that a configuration of polygons in the plane is guaranteed to remain stable.

Chapter 5 describes areas for further research. Appendix A contains definitions of polygons, configurations, and motion, as well as results used to show the complexity of stability problems. (The reader should refer to it as necessary while reading the thesis.) Appendix B contains an index of notation and terms.

1.2 Previous work

The field of mechanics, and more specifically, that of statics, has been studied for centuries by physicists and engineers. The theoretical foundations of statics are well established from both the standpoint of theoretical physics and in traditional areas of engineering practice. However, the computational aspects of testing stability remain largely uninvestigated.

In the early 1970's, Blum, Griffith, and Neumann developed a numerical method for stability testing[BGN70] at the MIT AI Lab. They used a rigid model of objects, which generally yields an underdetermined system of equations. The BGN algorithm modeled stability by assuming that a configuration is stable if it is possible to find an assignment of forces that satisfy the *force conditions for stability* (described in Section 4.2.5) with k of the variables set to zero, where k is the number of extra variables in the system of equations. Unfortunately such a condition does not guarantee stability, as the configuration in Figure 1.1 shows.

S. E. Fahlman, also at MIT, wrote a program for planning block constructions [Fa74]. His method constructed a data structure with symbolic representation of the configuration using "supported-by" assertions. When this was not sufficient, the system constructed the force equations and iterated to a solution.

No complexity analysis is included in either paper, with the exception that Fahlman noted that his method could fail to terminate.

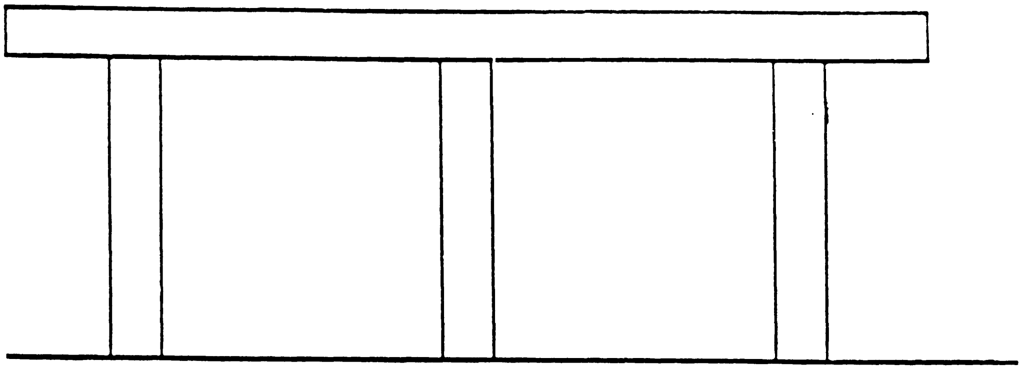

Figure 1.1: An unconditionally stable configuration.

1.3 Modeling rigid objects

The traditional approach to statics taken in mechanical engineering differs from that needed for a computational stability test. A mechanical engineer generally models physical objects as deforming in the presence of force. Most engineering statics problems arise in the analysis of large structures where it is necessary to view physical objects as deformable and represent reactive force as a continuous function over the surfaces of contact.

The domain of objects manipulated by a robot, however, is quite different. These objects tend to be relatively small and rigid, such as the hard plastic used in circuit boards or metal objects, such as nuts and bolts. A fundamental problem in modeling such objects is the inherent uncertainty in their dimensions and location. It is difficult to verify, for instance, that an assembly plan is correct without knowing the exact position and dimensions of the objects in the workcell.

With rigid bodies, any but the simplest configurations of physical objects subject to the force of gravity yield underdetermined systems of force and moment equations. Even if we assume that all bodies are polygons in the plane, and that force occurs only at the nodes of the polygons, almost all configurations are statically indeterminate. This makes it impossible to determine whether certain configurations, (the *potentially unstable* configurationss) are stable. Configurations of rigid polgyons are therefore either unconditionaly stable, unconditionally unstable, or neither. Figure 1.1 exhibits an unconditionally stable configuration, Figure 1.2 exhibits latter. Both configurations are statically indeterminate.

Using a deformable model guarantees a unique solution, but is unrealistic for most robotics applications because amount that a rigid object deforms due to loading is likely to be far less than the uncertainty in its dimensions and position. Figure 1.2 illustrates the problem. If the mass of the table top is supported entirely by the middle leg, then

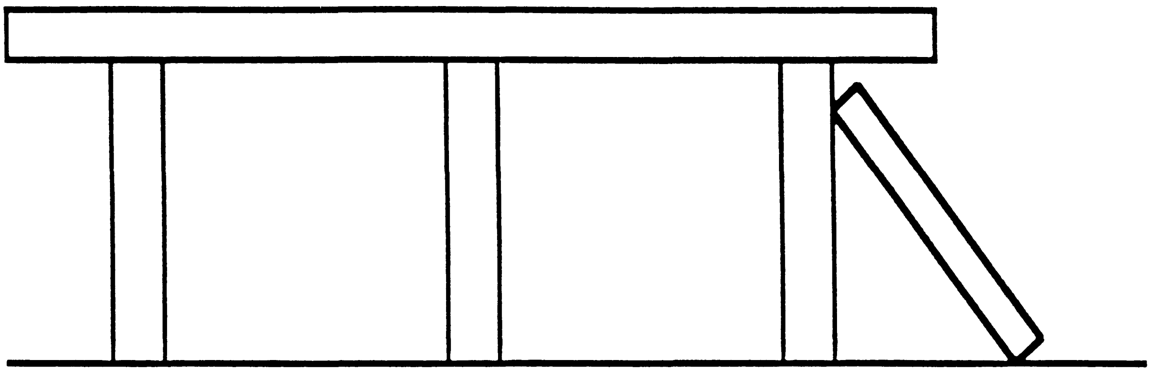

Figure 1.2: A *potentially unstable* configuration.

the plank can cause the right leg to slide. (If the coefficient of friction between the leg and floor is small enough.) If the mass of the table is supported by the outside legs, the configuration is stable. (If the plank is not too heavy.) Using a deformable model results in a unique solution, but not a useful one. The correct solution is given by the rigid model: it is impossible to guarantee that the configuration is stable.

Chapter 2

The Model

This chapter describes a simple model of oriented physical objects: configurations of polygons in the plane. It defines properties of the motion of configurations of polygons, which are used in subsequent chapters. Appendix A contains definitions and proofs used in this and subsequent chapters.

2.1 Polygons

A simple polgyon in the plane comprises an ordered set of edges that partition the the plane into three sets: the *interior*, the *boundary*, and the *exterior*. Appendix A formally defines simple polygons.

Let $P = P_1, \ldots, P_k$ be an ordered list of simple polygons.

2.1.1 Oriented polygons

Definition 2.1 An *oriented* polygon is a polygon P_i, together with an ordered triple $l_i = (x_i, y_i, \theta_i)$.

An oriented polygon P_i is defined in terms of its origin, or *reference frame*, and the triple l_i represents a *transformation* of the position of P_i. The oriented polygon (P_i, l_i) can be viewed as the result of

1. aligning P_i's reference frame with that of the plane,
2. rotating the polygon (and its reference frame) about the plane origin by θ_i, and
3. translating the polygon's reference frame to the point (x_i, y_i) on the plane.

2.1.2 Polygons with mass

Rigid physical objects with mass are modeled by assigning a mass, moment of inertia, and center of mass to each polygon.

Definition 2.2 A *polygon with mass* $(P_i, pm(i))$ is a polygon P_i, together with a tuple $pm(i) = (x, y, m, mi)$, with $m > 0$, $mi > 0$, which asserts that the center of mass is located at position (x, y) relative to P_i's reference frame, the mass of P_i is m, its moment of inertia about the center of mass is mi.

Let $mass(i) = m$, and $cm(i) = (x, y)$, and $I_{cm(i)} = mi$, for a polygon P_i with mass.

2.2 Configurations of polygons

Notation 2.1 *A* configuration C *of* P *is a sequence of ordered triples* $l_1, ..., l_n$, *where each* $l_i = (x_i, y_i, \theta_i)$ *represents the position (and orientation) of* P_i *in the plane.*

Notation 2.2 (P, C) *denotes the ordered list of polygons* P *in configuration* C. P *can be viewed as denoting the configuration* (P, C_z), *where* $C_z = ((0,0,0), \ldots, (0,0,0))$.

2.2.1 Feasible configurations

A *feasible* configuration contains no intersecting polygons. Such a configuration is called feasible because two physical objects can not occupy the same space.

Definition 2.3 A configuration C is *positionally feasible* for P if and only if the interiors of no distinct polygons P_i and P_j in P intersect.

Section A.1.1 of Appendix A defines the feasibility of configurations formally.

2.2.2 Contact

Two polygons P_i and P_j are *in contact* if they are in a positionally feasible configuration and the intersection of their boundaries is nonempty.

Definition 2.4 For a feasible configuration C, define $Contact(P_i, P_j)$ to be the intersection of the boundaries of P_i and P_j.

2.3 Motion

Definition 2.5 A *motion* $M(t)$ of a sequence of polygons P is a twice differentiable function of one variable that assigns, for each value of t, a position on the plane to each polygon P_i in P. That is, $M(t)$ is a vector of $M_i(t)$'s, where each $M_i(t) = (x_i(t), y_i(t), \theta_i(t))$ determines the position of polygon P_i for a given value of t.

Because the velocity and acceleration are modeled as continuous, each M_i is twice differentiable.

2.3.1 Translations

A *translation* is a "straight line" motion. In the context of the motion of polygons, a translation causes each point of a polygon to move in a straight line. (More generally, a translation M^i causes any point whose position is defined by M_i to move in a straight line.)

2.3.2 Feasible motions

Definition 2.6 $M(t) : [a, b]$ denotes the motion $M(t)$ restricted to the domain $a \le t \le b$.

Definition 2.7 A motion $M(t) : [a, b]$ is *feasible* for P if and only if the interiors of no distinct polygons P_i and P_j in P intersect for any value of t in the range $[a, b]$. Thus, a motion $M(t) : [a, b]$ is feasible for P if and only if $\{M(t) | a \le t \le b\} \subset Feasible(P)$.

Lemma 2.1 *If $M(t) : [a, b]$ is feasible for P, then there is a feasible motion $\widehat{M}(t) : [0, 1]$ for P. Thus we may assume without loss of generality that $a = 0$, and $b = 1$.*

Proof Let $\widehat{M}(t) = M(f(t))$, where $f(t) = (b - a)t + a$. ∎

2.3.3 Real motion

To be interesting, a motion $M(t) : [0, 1]$ must change the relative position of some polygons in P. A *real* motion changes the relative position of least one pair of polygons in P.

Definition 2.8 A motion $M(t) : [0, 1]$ is *real* if

$$(\exists i, j, a, b : 0 \le a < b \le 1)(M_i(a) - M_j(a) \ne M_i(b) - M_j(b)). \tag{2.1}$$

A configuration C of P is said to have a *real feasible motion* if there is a real feasible motion $M(t) : [0, 1]$, such that $M(0) = C$.

2.3.4 Path equivalent motion

Definition 2.9 Two motions $M(t)$ and $\widehat{M}(t)$ are *path equivalent* if there is a continuous nondecreasing function $f(t)$ such that $M(t) = \widehat{M}(f(t))$, $f(a) = a$, and $f(b) = b$.

The following lemma shows that we may assume that a real feasible motion has at least one polygon with a nonzero velocity at $t = 0$.

Lemma 2.2 *If a motion $M(t) : [0, 1]$ is real, then there is a path equivalent motion $\widetilde{M}(t) : [0, 1]$ such that some polygon P_i has a nonzero initial velocity, i.e., there is some i such that $\widetilde{M}_i'(0) \ne (0, 0, 0)$.*

The proof is in Section A.2.2 of Appendix A.

2.3.5 Motion of Points

Suppose the position of a point p is defined by M_i, the reference frame of P_i. (For instance, if p is a node of P_i.) Then the motion of p can be given as a function of M_i.

If $M_i(t) = (x_i(t), y_i(t), \theta_i(t))$ is the position of P_i, and (x_p, y_p) is the displacement of a point p in P_i's reference frame, the position of p at t is

$$\begin{aligned} Pos(p, M_i(t)) &= (-\sin(\theta_i(t))y_p + \cos(\theta_i(t))x_p + x_i(t), \\ &\quad \cos(\theta_i(t))y_p + \sin(\theta_i(t))x_p + y_i(t)). \end{aligned} \tag{2.2}$$

The homogeneous form of the motion $M_i(t) = (x_i(t), y_i(t), \theta_i(t))$, is the matrix

$$\begin{bmatrix} \cos\theta_i(t) & \sin\theta_i(t) & 0 \\ -\sin\theta_i(t) & \cos\theta_i(t) & 0 \\ x_i(t) & y_i(t) & 1 \end{bmatrix}. \tag{2.3}$$

$\mathrm{Pos}(p, M_i(t))$ can be represented as the matrix equation

$$\mathrm{Pos}(p, M_i(t) = \begin{matrix}[x_p, y_p, 1] \\ \\ \\ \end{matrix} \begin{bmatrix} \cos\theta_i(t) & \sin\theta_i(t) & 0 \\ -\sin\theta_i(t) & \cos\theta_i(t) & 0 \\ x_i(t) & y_i(t) & 1 \end{bmatrix}. \tag{2.4}$$

Since the columns of the homogeneous transformation defined by $M_i(t)$ are linearly independent, $M_i(t)$ has a unique inverse.

Notation 2.3 *Let $M_i^{-1}(t)$ denote the inverse of $M_i(t)$. Thus if $M_i(t)$ is viewed as a homogeneous transformation, i.e.,*

$$M_i(t) = \begin{bmatrix} \cos\theta_i(t) & \sin\theta_i(t) & 0 \\ -\sin\theta_i(t) & \cos\theta_i(t) & 0 \\ x_i(t) & y_i(t) & 1 \end{bmatrix},$$

then

$$M_i^{-1}(t) = \begin{bmatrix} \cos\theta_i(t) & -\sin\theta_i(t) & 0 \\ \sin\theta_i(t) & \cos\theta_i(t) & 0 \\ \widehat{x_i}(t) & \widehat{y_i}(t) & 1 \end{bmatrix}, \tag{2.5}$$

where

$$\widehat{x_i}(t) = -x_i(t)\cos\theta_i(t) - y_i(t)\sin\theta_i(t), \tag{2.6}$$

and

$$\widehat{y_i}(t) = x_i(t)\sin\theta_i(t) - y_i(t)\cos\theta_i(t) \tag{2.7}$$

Velocity at a point

If (x_p, y_p) is the displacement of a point p in P_i's reference frame, the velocity of p at t is given by

$$\begin{aligned} Pos'(p, M_i) \;=\; & (-\cos(\theta_i(t))\theta_i'(t)y_p - \sin(\theta_i(t))\theta_i'(t)x_p \\ & +x_i'(t), \\ & -\sin(\theta_i(t))\theta_i'(t)y_p + \cos(\theta_i(t))\theta_i'(t)x_p \\ & +y_i'(t)). \end{aligned} \tag{2.8}$$

The homogeneous representation of the velocity at a point $p = (x, y)$ is given by

$$Pos'(p, M_i) = [x, y, 1] \begin{bmatrix} -\sin\theta_i(t)\theta_i'(t) & \cos\theta_i(t)\theta_i'(t) & 0 \\ -\cos\theta_i(t)\theta_i'(t) & -\sin\theta_i(t)\ \theta_i'(t) & 0 \\ x_i'(t) & y_i'(t) & 1 \end{bmatrix}. \tag{2.9}$$

2.3.6 Point motion when $\theta_i(t_0) = 0$

If $p = (x_p, y_p)$, and $M_i(t)$ is such that $\theta_i(t) = 0$, for $t = t_0$, then

$$\mathrm{Pos}(p, M_i(t_0)) = (x_i(t_0) + x_p, y_i(t_0) + y_p). \tag{2.10}$$

The velocity at p is given by

$$V_p = (x_i'(t_0) - y_p\theta_i'(t_0),\ y_i'(t_0) + x_p\theta_i'(t_0)). \tag{2.11}$$

2.3.7 Relative motion

Definition 2.10 Suppose $M(t) = (M_1(t), \ldots, M_k(t))$ is a motion of P, and $M_r(t) = (x_r(t), y_r(t), \theta_r(t))$. The *relative motion*

$$M^r(t) = (M_1{}^r(t), \ldots, M_k{}^r(t)),$$

where

$$M_i{}^r(t) = M_i(t)M_r{}^{-1}(t).$$

$M^r(t)$ gives the positions of the polygons of P relative to $M_r(t)$.

Lemma 2.3 *For any M_r, M^r is equivalent to M in the sense that*

$$Pos(p_1, M_i) = Pos(p_2, M_j)$$

if and only if

$$Pos(p_1, M_i{}^r) = Pos(p_2, M_j{}^r).$$

Proof Since

$$\mathrm{Pos}(p_1, M_i) = p_1 M_i = \mathrm{Pos}(p_2, M_j) = p_2 M_j, \tag{2.12}$$

it follows that

$$p_1 M_i {M_r}^{-1} = p_2 M_j {M_r}^{-1}, \tag{2.13}$$

so

$$p_1 {M_i}^r = p_2 {M_j}^r, \tag{2.14}$$

and

$$\mathrm{Pos}(p_1, {M_i}^r) = \mathrm{Pos}(p_2, {M_j}^r). \tag{2.15}$$

∎

Corollary 2.1 *A motion $M(t) : [0,1]$ is real and feasible for P if and only if M^r is real and feasible for any M_r.*

The following lemma shows that we may assume that, for a given motion $M(t) : [0,1]$ of P, the polygons of P are aligned with the plane origin at $t = 0$

Lemma 2.4 *Let $M(t)$ be any motion of P. Then there is an equivalent motion $\widehat{M}$ and sequence of polygons $\hat{P}$, with the property that $\widehat{M}(0) = ((0,0,0), \ldots, (0,0,0))$.*

Proof For each polygon P_i in P, define

$$\hat{P}_i = (\hat{n}_{i1}, \ldots, \hat{n}_{ik_i}),$$

where $\hat{n}_{ij} = Pos(n_{ij}, M_i(0))$. Let $\hat{P} = (\hat{P}_1, \ldots \hat{P}_k)$. Let $\widehat{M} = (\widehat{M}_1, \ldots \widehat{M}_k)$, where $\widehat{M}_i(t) = M_i(t){M_i}^{-1}(0)$. Then $\widehat{M}(0) = ((0,0,0), \ldots, (0,0,0))$, and $(\hat{P}, \widehat{M})$ is equivalent to (P, M) in the sense that the position of each polygon $\hat{P}_i$ in the plane is identical to that of P_i, for every value of t. ∎

satisfies $M(0) = ((0,0,0), \ldots, (0,0,0))$.

Definition 2.11 A motion M is *initially zero* if

$$M(0) = ((0,0,0), \ldots, (0,0,0)).$$

Chapter 3

Mobility

Mobility is the problem of determining whether there is a simultaneous non-intersecting motion of a given set of oriented polygons. The mobility problem is of considerable interest in formulating a stability algorithm since a configuration is guaranteed to be stable if there is no feasible simultaneous motion of its polygons. This chapter defines and shows the complexity of several mobility problems, which are stated informally below:

1. Infinitesimal mobility: Is there an infinitesimally feasible velocity for a given configuration of polygons P? (Infinitesimal feasibility is defined in Section 3.2.)
2. Translational mobility: Is there a real feasible translational motion for a given configuration P?
3. General mobility: Is there a real feasible motion for a given configuration P?

Infinitesimal and translational mobility are NP-complete. General mobility is NP-hard, but no upperbound is known.

3.1 Velocity constraints on feasible motions

This section develops constraints on the velocity of feasible motions, which are used to show that infinitesimal mobility is in NP. Section 3.1.1 develops constraints that a feasible motion of a point p satisfies. Using the point constraints of Section 3.1.1, Section 3.1.2 derives constraints on the relative motion of contacting polygons. Section 3.1.2 combines the constraints of Section 3.1.2 to get constraints on real feasible motions of a configuration P.

3.1.1 Feasibility constraints on the velocity of a point

This section describes constraints on the motion of a point in contact with a polygon in the plane. Suppose that $M_p(t)$ is a motion function for a point p, and that p lies on

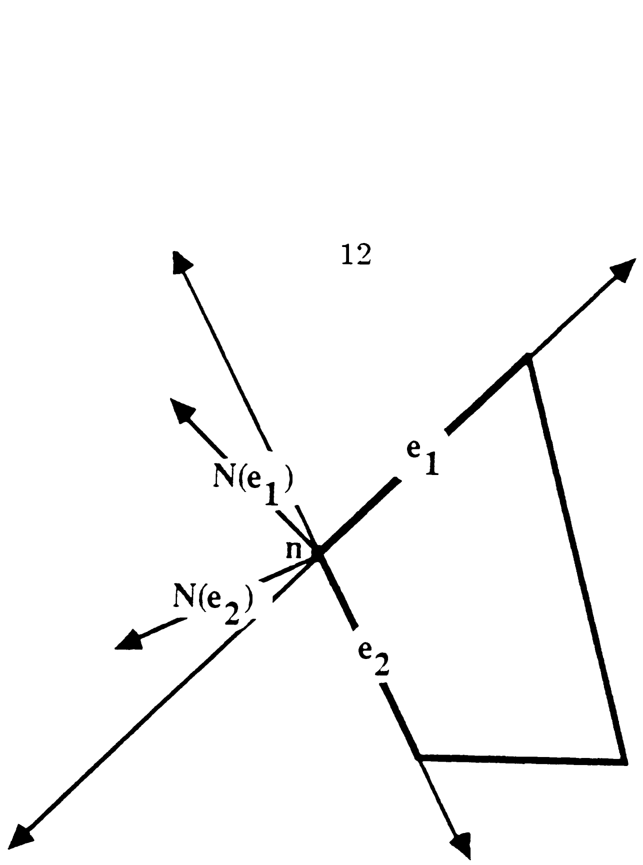

Figure 3.1: A point in contact with a node forming an interior angle $\phi < \pi$

the boundary of a polygon P_i at $t = 0$. P_i is assumed to be fixed in the plane in this section, and the problem considered is to define constraints that guarantee that there is a feasible motion of p. (A feasible motion of p with respect to P_i is one for which p is not in the interior of P_i for any value of t.)

Note that although the constraints developed in the section will be used later to construct feasibility constraints for polygons and configurations of polygons, this section formulates constraints that all feasible motions of a single point satisfy.

Suppose that a point p is in contact with a polygon P_i at $t = 0$ (i.e., $M_p(0) \cap Boundary(P_i) = M_p(0)$), and that $M_p(t) : [0,1]$ is a feasible motion of p. Let $V_p = M_p'(0) = (x, y)$ be the initial velocity of p. By Lemma A.1 we may assume that p coincides with a node of P_i. Assume without loss of generality that p coincides with $n = e_1 \cap e_2$.

The feasibility constraints for V_p depend on the interior angle formed at n. There are three cases: when the interior angle formed at n is less than, equal to, and greater than π.

1. If the interior angle ϕ formed by e_1 and e_2 at n is less than π, as shown in Figure 3.1, then $V_p \cdot N(e_1) \geq 0$ or $V_p \cdot N(e_2) \geq 0$.[1]

2. If $\phi = \pi$ then $V_p \cdot N(e_1) \geq 0$. Figure 3.2 illustrates point contact with a polygon at a node that forms an angle of π.

3. If $\phi > \pi$, then $V_p \cdot N(e_1) \geq 0$ and $V_p \cdot N(e_2) \geq 0$. Figure 3.3 illustrates point contact with a node of a polygon that forms an interior angle less than π.

[1]The symbol "·" denotes the scalar product of vectors

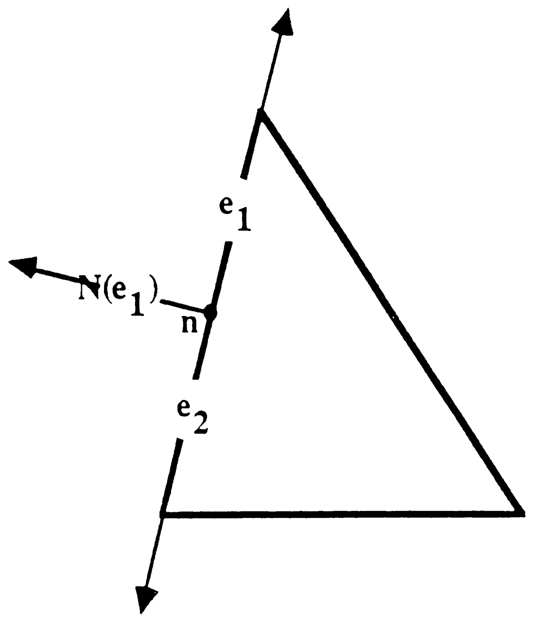

Figure 3.2: Point contact with a node forming an interior angle $\phi = \pi$

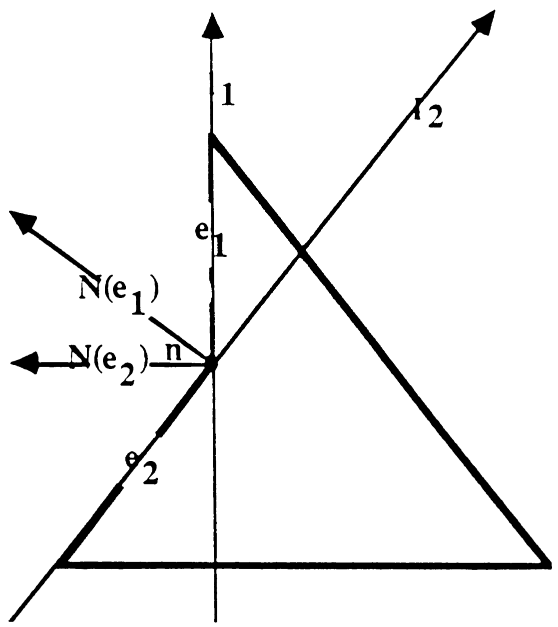

Figure 3.3: Point contact with a node forming an interior angle $\phi > \pi$

These constraints are formulated in the following definition.

Definition 3.1 Define

$$\mathrm{DirExt}(p, P_j, V_p) = \begin{cases} (V_p \cdot N(e_1) \geq 0) \vee (V_p \cdot N(e_2) \geq 0) & \text{if } \phi < \pi \\ (V_p \cdot N(e_2) \geq 0) & \text{if } \phi = \pi \\ (V_p \cdot N(e_1) \geq 0) \wedge (V_p \cdot N(e_2) \geq 0) & \text{if } \phi > \pi \end{cases}.$$

$\mathrm{DirExt}(p, P_j, V_p)$ asserts that the velocity of p is not directed towards the interior of P_j in the neighborhood of p. Usually p will be a point in $Contact(P_i, P_j)$, for some polygon P_i, and V_p will be the velocity of p relative to P_j.

Theorem 3.1 (Point velocity constraints) *If V_p is a feasible motion of p with respect to P_j $DirExt(p, P_j, V_p)$ is satisfied.*

Proof

1. Suppose that the antecedent of Case 1 is satisfied, i.e., that p is coincident with a node of P_i that forms an interior angle less than π. Suppose that V_p is the velocity of some feasible motion $M_p(t)$ of p, i.e., $V_p = M_p'(0)$ for some real feasible $M_p(t) : [0, 1]$. Suppose, for the purpose of contradiction, that $V_p \cdot N(e_1) < 0$ and $V_p \cdot N(e_2) < 0$. Let l_1 and l_2 be lines passing through n that are parallel to e_1 and e_2, respectively, as illustrated in Figure 3.1. Then by elementary calculus, we can find $\epsilon_1 > 0$ sufficiently small such that the position of p, $M_p(t)$, is such that for $0 < t < \epsilon_1$, $(M_p(t) - M_p(0)) \cdot N(e_1) < 0$. Thus $M_p(t)$ and $M_p(0) + N(e_1)$ are on opposite sides of l_1 for $0 < t < \epsilon_1$. Similarly, we can find $\epsilon_2 > 0$ such that $0 < t < \epsilon_2 \Rightarrow (M_p(t) - M_p(0)) \cdot N(e_2) < 0$, so $M_p(t)$ and $M_p(0) + N(e_2)$ are on opposite sides of l_2 for $0 < t < \epsilon_2$. Therefore, for all $0 < t < min\{\epsilon_1, \epsilon_2\}$, $M_p(t)$ is not feasible for p, a contradiction.

2. If $\phi = \pi$, the proof is similar to that of Case 1. In this case $N(e_1) \cdot V_p < 0 \iff N(e_2) \cdot V_p < 0$, and V_p is not feasible.

3. If $\phi > \pi$, the proof is also similar to that for Case 1, except that in this case if either of the scalar products $V_p \cdot N(e_1)$, or $V_p \cdot N(e_2)$, is negative, then p intersects the interior of P_i for sufficiently small $\epsilon > 0$. This follows from the fact that if p is on the "wrong" side of either l_1 or l_2, then it is in the interior of P_i for sufficiently small ϵ.

∎

The triple (p, P, V) is said to satisfy Theorem 3.1 if $\mathrm{DirExt}(p, P, V)$ is true.

3.1.2 Feasibility constraints on the relative velocity of two polygons in contact

This section derives conditions that the velocity of any feasible relative motion of two polygons in contact must satisfy. Section 3.1.2 will show that any real feasible motion $M(t) : [0, 1]$ of $P = (P_1, \ldots, P_n)$ must satisfy the conditions derived in this section for each pair of polygons in contact at $t = 0$.

Node constraints

Section 3.1.1 showed that if a point p is in contact with a polygon P_j at $t = 0$, and $M_p(t) : [0,1]$ is feasible then DirExt(p, P_j, V_p) is satisfied. If two polygons P_i and P_j are in contact at $t = 0$, then the relative velocity of any point $p \in Contact(P_i, P_j)$ must satisfy both DirExt(p, P_j, V_{pi}) and DirExt(p, P_i, V_{pj}), where V_{pi} is the relative velocity of p with respect to P_j when p is viewed as a point of P_i, and V_{pj} is the relative velocity of p with respect to P_i when p is viewed as a point of P_j.

Theorem 3.2 *Suppose a relative motion of $M_i{}^j : [0,1]$ is feasible for P_i and P_j, and suppose $p_i \in Contact(P_i, P_j)$ at $t = 0$. Then $(p_i, P_j, M_p{}^j)$ satisfies Theorem 3.1.*

Proof Otherwise $Pos(p_i, M_j(\epsilon)) \cap Interior(P_j) \neq \emptyset$, for $\epsilon > 0$ sufficiently small. ▮

Thus if a relative motion of two polygons is feasible, Theorem 3.1 is satisfied for every point of contact.WR

Definition 3.2 [Infinitesimal feasibility] A motion M is *infinitesimally feasible* for P at t if the relative velocity of every point $p \in Contact(P_i, P_j)$, for any distinct pair of polygons P_i and P_j satisfies Theorem 3.1. More generally, if S is any set of points, a motion M is said to be infinitesimally feasible for S if Theorem 3.1 is satisfied for all $p \in S$, that is if the relative velocity of every point $p \in S$ satisfies Theorem 3.1. Thus to assert that a motion is infinitesimally feasible is to assert that the motion can not be shown to be infeasible because of its velocity.

A velocity V is *infinitesimally feasible* if it is the velocity of an infinitesimally feasible motion M.

Edge constraints

The following lemma shows that any non-translational velocity V (i.e., $V \neq (x, y, 0)$) is the velocity of a constant rate (pure) rotation about some point in the plane. This is used to derive constraints for the feasibility of motions with edges of contact.

Lemma 3.1 *Suppose $V = (x, y, \omega)$, is the velocity of some motion function M_i, such that $\omega \neq 0$. Then V is the velocity of some $\widehat{M_i}(t)$, such that $\widehat{M_i}(t)$ is a rotation about some point (x_v, y_v).*

Proof By the definition of the velocity of a point (Equation 2.11), the point $p_z = (x_v, y_v) = (y/\omega, -x/\omega)$ has velocity $(0,0)$, since

$$((-y\omega)/\omega + y, (-x\omega)/\omega + x) = (0,0). \tag{3.1}$$

Define

$$\begin{aligned}\widehat{M_i}(t) \quad = \quad & (x_v(1 - cos(\omega t)) - y_v \sin(\omega t), \\ & y_v(\cos(\omega t) - 1) - x_v \sin(\omega t), \\ & \omega t).\end{aligned}$$

$\widehat{M_i}(t)$ is the transformation that rotates the plane about the pointp_z by ωt. This can be seen by noticing that it is the result of translating to align p_z with the origin, rotating ωt, and translating the origin back to p_z:

$$\mathcal{H}(\widehat{M_i}(t)) = \begin{bmatrix} \cos 0 & \sin 0 & 0 \\ -\sin 0 & \cos 0 & 0 \\ x_v & y_v & 1 \end{bmatrix} \tag{3.2}$$

$$= \begin{bmatrix} \cos,\omega t & \sin,\omega t & 0 \\ -\sin,\omega t & \cos,\omega t & 0 \\ \widehat{x}(t)\omega t & \widehat{y}(t)\omega t & 1 \end{bmatrix}, \tag{3.3}$$

where

$$\widehat{x}(t) = x_v(1 - cos(\omega t)) - y_v \sin(\omega t), \tag{3.4}$$

and

$$\widehat{y}(t) = y_v(\cos(\omega t) - 1) - x_v \sin(\omega t). \tag{3.5}$$

The velocity of $\widehat{M_i}(t)$ is then

$$\begin{aligned} \widehat{M_i}'(t) &= \frac{d}{dt}(x_v(1 - cos(\omega t)) - y_v \sin(\omega t), & (3.6)\\ &\qquad y_v(\cos(\omega t) - 1) - x_v \sin(\omega t), \omega t) & (3.7)\\ &= (x_v(\sin \omega t) - y_v \cos(\omega t))\omega t', & (3.8)\\ &\qquad (-y_v(\sin \omega t) - x_v \cos(\omega t))\omega t', \omega t') & (3.9)\\ &= (x_v(\sin(\omega t) - y_v \cos(\omega t))\omega, & (3.10)\\ &\qquad -y_v(\sin(\omega t) - x_v \cos(\omega t))\omega, \omega). & (3.11) \end{aligned}$$

At $t = 0$, this becomes

$$\begin{aligned} \widehat{M_i}'(t) &= (-y_v\omega, -x_v\omega, \omega) & (3.12)\\ &= (-(-x/\omega)\omega, -(y/\omega)\omega, \omega) & (3.13)\\ &= (x, y, \omega). & (3.14) \end{aligned}$$

Thus every velocity transformation is the velocity of a rotation about some point in the plane. ∎

Lemma 3.1 is used in the following theorem, which shows that it is possible to determine whether or not every point along an edge of contact satisfies Theorem 3.1 by checking constraints only at the endpoints. This allows the feasibility of the velocity of all the points of e to be tested in finite time, as is necessary for an algorithm to check for infinitesimal feasibility.

Theorem 3.3 (Edge constraints on velocity) *Suppose $M(t)$: $[0,1]$ is feasible for P, and that a line segment l is such that $l \in Contact(P_i, P_j)$ at $t = 0$. (That is, l is a line segment of contact between P_i and P_j.) Without loss of generality (by Lemma A.1), suppose that l coincides with e_i and e_j, edges of P_i and P_j, respectively. Let $e_i = (n_{i_1}, n_{i_2})$,*

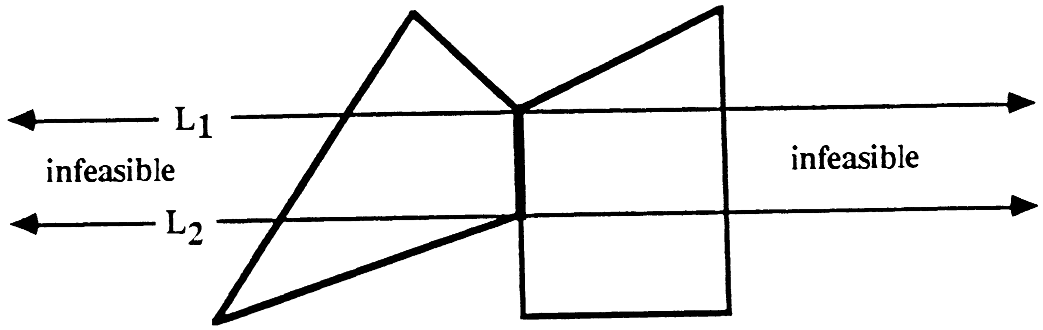

Figure 3.4: Points of feasible rotation

and let V_1 and V_2 be the velocities of n_{i_1} and n_{i_2}, relative to that of P_j. Then $V = M'(0)$ is infinitesimally feasible for $p \in e_i$ (with respect to e_j) if and only if $V_1 \cdot N(e_j) \geq 0$ and $V_2 \cdot N(e_j) \geq 0$. Formally,

$$((\forall p \in e_i)(V_p \cdot N(e_j) \geq 0)) \iff (V_1 \cdot N(e_j) \geq 0 \wedge V_2 \cdot N(e_j) \geq 0).$$

Proof

($\Rightarrow$) Suppose, to the contrary, that (without loss of generality) $V_1 \cdot N(e_j) < 0$. Then regardless of the value of V_2, we can find $p \in e_i$ sufficiently close to n_{i_1} such that $V_p \cdot N(e_j) < 0$, where V_p is the velocity of the point p. Thus by elementary calculus M_i is not feasible for p, a contradiction.

($\Leftarrow$) Suppose that $V_1 \cdot N(e_j) \geq 0$, and $V_2 \cdot N(e_j) \geq 0$. There are two cases:

1. If $V = (x, y, 0)$, (i.e., V is a translational velocity), then $V_p = (x, y)$, for all p. It then follows that $V_p \cdot N(e_j) = V_1 \cdot N(e_j) \geq 0$. Therefore V satisfies Case 2 of Theorem 3.1.

2. If $V = (x, y, \omega)$, where $\omega \neq 0$, then by Lemma 3.1, V is an infinitesimal rotation about some point. Consider Figure 3.4. L_1 and L_2 are the lines parallel to $N(e_j)$ that pass through n_{j_1}, and n_{j_2}, respectively. Since both $V_1 \cdot N(e_j) \geq 0$, and $V_2 \cdot N(e_j) \geq 0$, it must be the case that the point of zero rotation is not in the the region defined by the intersection of open half-planes on the "e_j" sides of L_1 and L_2 (labeled "infeasible" in Figure 3.4). To see this, suppose that the point of zero rotation is between L_1 and L_2. If the rotation is clockwise, then $V_1 \cdot N(e_j) < 0$, a contradiction. If the rotation is counterclockwise, then $V_2 \cdot N(e_j) < 0$, which is also a contradiction.

   Thus the point of zero rotation must be either on or above L_1, in which case it must be clockwise, or on or below L_2, in which case it is counterclockwise. It is straightforward to verify that, for any $p \in e_i$, $V_p \cdot N(e_j) \geq 0$, in either of these cases.

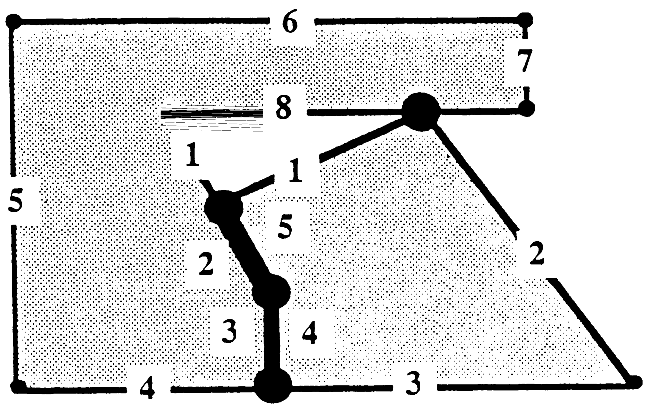

Figure 3.5: Contact regions for two polygons

∎

Polygon constraints

Section 3.1.1 derived conditions that the velocity of any point p in contact with a polygon P_j must satisfy, and this section derived feasibility conditions on the velocity of points on an edge of contact. Since every point of contact between two polygons is either at a node or on an edge of contact, and there are a finite number of points and edges, it is possible to construct a finite formula that is **TRUE** if and only if the relative velocities of the polygons is infinitesimally feasible.

For a given pair of polygons P_i and P_j, $Contact(P_i, P_j)$ can be decomposed into disjoint connected regions

$$R = (r_1, \dots, r_k),$$

where

$$Contact(P_i, P_j) = r_1 \cup \dots \cup r_k,$$

where each $r_j = (n_1, \dots, n_{j_k})$, and each n_i is a node of either P_i or P_j. Figure 3.5 shows an example of two such regions. In this case

$$Contact(P_i, P_j) = r_1 \cup r_2,$$

where r_1 comprises the edges e_4 and e_5 of P_j, and r_2 is node n_2 of P_j.

Definition 3.3 Suppose a point $p \in Contact(P_i, P_j)$ in configuration P, and that the position of $p = (x_p, y_p)$ is defined with respect to P_i's reference frame. Let $V = (V_1, \dots, V_n)$ be a vector of velocities, where V_i is the velocity of P_i. Define the *point constraint* $PtC(p, P_j, V)$ to be that case of DirExt(p, P_j, V_i^j) that applies to p and P_j, where V_i^j is the relative velocity of P_i with respect to that of P_j.

Suppose p is a point on the boundary of P_i . Then $PtC(p, P_j, V)$ constrains V to be infinitesimally feasible for the point p with respect to P_j.

Definition 3.4 Suppose $e = (n_1, n_2)$ is an edge of P_i in contact with P_j, and $n_1 = (x_1, y_1)$ and $n_2 = (x_2, y_2)$. Define the *edge constraint*

$$EC(e, V) = V_{n_1} \cdot -N(e) \geq 0 \wedge V_{n_2} \cdot -N(e) \geq 0,$$

where V_{n_i} is the relative velocity of n_i with respect to that of P_j. The *edge constraint* $EC(e, V)$ constrains the velocity to be infinitesimally feasible for the edge of contact e.

Definition 3.5 For each $r_j \in R$, define the *region constraints* to be

$$\begin{aligned} RC(r_j, V) &= \bigwedge_{1 \leq i \leq j_k} (PtC(n_i, P_j, V) \wedge PtC(n_i, P_i, V)) \\ &\wedge \bigwedge_{1 \leq i < j_k} EC((n_i, n_{i+1}), P_j). \end{aligned}$$

Definition 3.6 Define the *polygon constraints* on P_i and P_j to be

$$PgC(P_i, P_j, V) = \bigwedge_{r_j \in Contact(P_i, P_j)} RC(r_j, V).$$

Theorem 3.4 *The relative velocity of two polygons P_i and P_j is infinitesimally feasible if and only if V satisfies $PgC(P_i, P_j, V)$.*

Proof The theorem follows directly from Theorem 3.1, Definition 3.3, Theorem 3.3, Definition 3.4, and the definition of infinitesimal feasibility (Definition 3.2). ∎

Thus it is possible to verify that the relative velocity of every point of contact between two polygons is infinitesimally feasible by considering a finite number of constraints. (The number of constraints is linear in the number of nodes of the polygons.)

Configuration constraints

Since a velocity V of a configuration P is infinitesimally feasible if and only if the relative velocities of each pair of polygons in contact is, the constraints for an infinitesimally feasible velocity are as follows.

Definition 3.7 Define the feasibility constraints for the velocity of a configuration of polygons (P, C) to be

$$Constraints(P, C, V) = \bigwedge_{1 \leq i < j \leq n} PgC(P_i, P_j, V).$$

$Constraints(P, C, V)$ is a formula containing base terms of the form $V_p \cdot N(e) \geq 0$, which are combined with the connectives **AND** and **OR**. V_p is a function of V and p. This formula will be used in Section 3.2 to show that the infinitesimal mobility problem is in NP.

3.2 Infinitesimal mobility

The results of Section 3.1.2 show that the initial velocity of any feasible motion $M(t)$: $[0,1]$ of P must satisfy certain feasibility constraints. This suggests the following problem:

Problem 3.1 (Infinitesimal mobility) *Is there a feasible velocity for a configuration of polygons $P = P_1, \ldots, P_n$, where each of the nodes of P is at a rational point.*

Every real feasible motion for a given configuration P has an initial velocity that is infinitesimally feasible. Thus the infinitesimal mobility problem can be viewed as an approximation to the general mobility problem. In particular, if there is no infinitesimally feasible velocity of a configuration, there is no real feasible motion for that configuration.

3.2.1 Infinitesimal mobility is in NP

This section shows that infinitesimal mobility (Problem 3.1) for configurations of polygons having nodes at rational points is in NP.

To show that infinitesimal mobility is in NP it suffices to exhibit a nondeterministic polynomial algorithm that determines whether a given configuration P has a feasible velocity V, i.e., to show that $V = (V_1, \ldots, V_n)$, satisfies Constraints(P, C) (the velocity constraints given in Section 3.1.2).

The following algorithm determines whether there is such a V.

Algorithm 3.1 [Infinitesimal mobility]

1. Construct the constraints for P.
2. Guess a velocity V.
3. Verify that V satisfies the constraints.

If each step of the algorithm can be accomplished in polynomial time, infinitesimal mobility is in NP. Note that the constraint set for P as well as the guessed velocity V must have representations that have size polynomial in the size of P. Thus it is necessary to show that if any infinitesimally feasible velocity exists, there is one with a representation that is small enough (i.e., one with size polynomial in the size of P).

Representing a configuration of polygons

Since the nodes of a polygons P_i are at rational points, we may assume that each P_i is of the form $P_i = (n_{i1}, \ldots, n_{ik})$, where each n_{ij} is an ordered pair of integers (a_{ij}, b_{ij}) that represents the point a_{ij}/b_{ij}.

Each rational number $q = a/b$ can thus be represented with $\lg(a) + \lg(b)$ bits. Let $Size(q)$ denote the number of bits in the representation of q. Then

$$\begin{aligned} Size(q_1 + q_2) &\leq \max\{Size(q_1), Size(q_2)\} + 1 && (3.15)\\ Size(q_1 q_2) &\leq Size(q_1) + Size(q_2). && (3.16) \end{aligned}$$

Let $s = \max\{Size(n_{ij})\}$, for n_{ij} in P. Thus, we may assume that the representation of each node n_{ij} requires at most s bits. For simplicity, let us assume that each node n_{ij} requires exactly s bits. This assumption is justified by the following lemma:

Lemma 3.2 *Suppose* $s = \max\{Size(n_{ij})\}$. *Then* $s \overset{P}{\preceq} Size(P) \overset{P}{\preceq} s$.

Proof Suppose r is the number of nodes in P. Then the representation of P requires at least r bits to represent this fact (i.e., the representations of the nodes must be separated). Thus $s + r \leq Size(P)$, and therefore $s \leq Size(P) \leq Size(P)s$. Thus, setting $f_1(s) = s$, and $f_2(s) = Size(P)s$, $f_1(s) \leq Size(P)$, and $Size(P) \leq s$, so $s \overset{P}{\preceq} Size(P) \overset{P}{\preceq} s$. ∎

Since all nodes are at rational points, we may assume by Lemma 2.4 that

$$C = ((0,0,0), \ldots, (0,0,0)),$$

i.e., that the reference frame of each polygon P_i is coincident with the plane origin.

Constraints

Each constraint is imposed by some point (or line segment) of contact between two polygons P_i and P_j. Since there are $(O(n^2))$ combinations of two polygons, the number of pairs of contacting polygons (P_i, P_j) such that $Contact(P_i, P_j) \neq \emptyset$ is bounded by a polynomial. $Contact(P_i, P_j)$ can be partitioned into a set of regions of contact, where each region is either a point or a line segment. The number of regions in this set is polynomial in the number of nodes in P_i and P_j. For each region, there is either one or two constraints on the velocity. Thus there are a polynomial number of constraints.

Form of constraints

This section shows that each base constraint is a linear inequality, and that the constraints can be represented in polynomial space.

Recall that each base constraint is of the form $N(e_i) \cdot V_p \geq 0$, where $N(e_i)$ is a vector normal to the edge e_i, and V_p is the relative velocity of the point p with respect to the velocity of e_i.

Let $C_{(e_i,p)}$ denote $N(e_i) \cdot V_p$.

Suppose that $e_i = (n_1, n_2) = ((x_1, y_1), (x_2, y_2))$, $p = (x_p, y_p)$, p is a point of P_j, and e_i is an edge of P_i. Then $N(e_i) = (y_2 - y_1, x_2 - x_1)$ (Definition A.5).

The relative velocity of p with respect to P_i, V_p, is given by Equation A.11, i.e.,

$$V_p = (x_j - x_i - y_p(\omega_j - \omega_i), y_j - y_i + x_p(\omega_j - \omega_i)). \tag{3.17}$$

Therefore,

$$\begin{aligned}
C_{(e_i,p)} &= N(e_i) \cdot V_p && (3.18)\\
&= (y_2 - y_1, x_2 - x_1) \cdot \\
&\quad (x_j - x_i - y_p(\omega_j - \omega_i), y_j - y_i + x_p(\omega_j - \omega_i)) && (3.19)\\
&= (y_2 - y_1)(x_j - x_i - y_p(\omega_j - \omega_i)) \\
&\quad +(x_2 - x_1)(y_j - y_i + x_p(\omega_j - \omega_j)) && (3.20)\\
&= (y_2 - y_1)(x_j - x_i) + (x_2 - x_1)(y_j - y_i) \\
&\quad +[(x_2 - x_1)x_p - (y_2 - y_1)y_p](\omega_j - \omega_i) && (3.21)\\
&= c_1(x_j - x_i) + c_2(y_j - y_i) + c_3(\omega_j - \omega_i), && (3.22)
\end{aligned}$$

where c_1, c_2, and c_3 are constants.

Thus $C_{(e_i,p)} \geq 0$, is a linear inequality. It is easy to verify that the coefficients c_1, c_2, and c_3 can be represented with no more than $s+1$, $s+1$, and $2s+3$ bits, respectively.

Constructing the constraints

Constructing the constraints requires that the algorithm identify the contact regions described in Section 3.1.2. Each of these regions is either:

1. A node of some polygon P_i intersecting a node or edge of a second (distinct) polygon P_j
2. The intersection of an edge of P_i with an edge of P_j.

This section describes a method to find these regions.

1. Determining that two nodes coincide is trivial, since it suffices to note that they have the same position.
2. Determining that a node lies on an edge is similarly straightforward, since a node n lies on an edge $e = (n_1, n_2)$ if and only if n satisfies the equation of the line passing through e, $(n - n_1) \cdot e \geq 0$, and $(n - n_2) \cdot \leq < 0$.
3. Determining that two edges e_1 and e_2 intersect can be done in a similar manner. The lines passing through e_1 and e_2 must be the same and a node (or both nodes) of one must satisfy the condition for Case 2. It is straightforward to determine the nature of the intersection between the edges (e.g., that e_2 is completely contained in e_1).

Thus it is possible for a polynomial time algorithm to determine the regions of contact and construct the feasibility constraints.

3.2.2 Guessing a velocity

If an algorithm for infinitesimal mobility is to "guess" a velocity V, V must have size polynomial in $Size(P)$. This section shows that if there is any velocity that satisfies Constraints(P), there is one with rational components and size polynomial in $Size(P)$.

Verifying that V satisfies Constraints(P)

Lemma 3.3 *Suppose V satisfies Constraints(P). Then there is a velocity $\widehat{V}$ that satisfies Constraints(P), and is such that $Size(\widehat{V}) \overset{P}{\preceq} Size(P)$.*

Proof
Suppose V satisfies Constraints(P). Then Constraints(P) satisfies the definition of F in Lemma A.4. Let $Z = (x_1, y_1, \omega_1, \ldots, x_n, y_n, \omega_n)$. By Lemma A.4, determining whether the triple $(Z, \text{Constraints}(P), 0)$ has a solution X is in NP. By hypothesis there is a solution to Constraints(P). Therefore, there is an X such that $Size(X) \overset{P}{\preceq} Size(P)$. Setting $\widehat{V} = X$ satisfies the statement of the lemma. ∎

Theorem 3.5 *Infinitesimal mobility is in NP.*

Proof The results of this section. ∎

This section has shown that the problem of determining whether there is an infinitesimally feasible velocity for a given configuration of polygons can be solved in nondeterministic polynomial time. Since any real feasible motion has an infinitesimally feasible velocity, this shows that the complement of the mobility problem is in NP, and thus that the mobility problem is in CO-NP.

3.3 Translational mobility

The translational mobility problem is determining whether a given configuration P has a feasible translational motion. Section 3.3.1 shows that translational mobility is in NP, while Section 3.3.2 shows that it is NP-hard. Thus translational mobility is NP-complete.

Problem 3.2 (Translational mobility) *Is there a feasible translational motion $M(t)$: $[0, 1]$ for a given configuration P?*

3.3.1 Translational mobility is in NP

This section shows that translational mobility is in NP by showing that a configuration P has a real feasible translational motion if and only if it has a feasible velocity that is a translation. Thus it is possible to determine that a given configuration P has a real feasible translation by guessing a translational velocity, and verifying that it is infinitesimally feasible as described in Section 3.2.

Lemma 3.4 *If $M_i(t)$ and $M_j(t)$ are constant rate translations such that $M_i(t) = V_i t$, and $M_j(t) = V_j t$, then $M_i{}^j$ is also a constant rate translation.*

Proof Recall that $M_i{}^j = M_i M_j^{-1}$. If M_i and M_j are constant rate translations, then

$$M_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_i(t) & y_i(t) & 1 \end{bmatrix}, \tag{3.23}$$

$$M_j = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_j(t) & y_j(t) & 1 \end{bmatrix}. \tag{3.24}$$

Thus

$$M_j^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_j(t) & -y_j(t) & 1 \end{bmatrix}, \tag{3.25}$$

and

$$M_i{}^j = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_i(t) - x_j(t) & y_i(t) - y_j(t) & 1 \end{bmatrix}, \tag{3.26}$$

which is a constant rate translation. ∎

Theorem 3.6 *Suppose there is a velocity V that is infinitesimally feasible for P, and that V is a translational velocity. Then there is a feasible motion $M(t) : [0, 1]$ for P.*

Proof Suppose that a translational velocity V is infinitesimally feasible for P. Define $M(t) = Vt$. Suppose that p, a point in $Boundary(P_i)$ is in $Contact(P_i, P_j)$. The fact that V is infinitesimally feasible for P guarantees that V_p, the relative velocity of p with respect to P_j, is infinitesimally feasible for p (i.e., satisfies Theorem 3.1). Let e_1 and e_2 be the the edges of P_j as in Theorem 3.1, and let $M_p(t)$ be the relative motion of p with respect to P_j. Since $M_p(t) - M_p(0)$ is parallel to and directed in the same direction as V_p for $t > 0$, it follows by Theorem 3.1 that p does not cross e_1 or e_2 to $Interior(P_j)$ for any $t > 0$. Since e_1 and e_2 are the only edges that p is in contact with (since P_j is simple), it follows that there is some $\epsilon_{p,P_j} > 0$ such that

$$(\forall t : 0 \leq t \leq \epsilon_{p,P_j})(Pos(p, M_i{}^j(t)) \Rightarrow p \notin Interior(P_j)).$$

That is, p is not in the interior for P_j for any value of t in the range $[0, \epsilon_{p,P_j}]$.

If $p \in Boundary(P_i)$ is not in $Contact(P_i, P_j)$ then there is also $\epsilon_{p,P_j} > 0$ such that

$$(\forall t : 0 \leq t \leq \epsilon_{p,P_j})(Pos(p, M_i{}^j(t)) \Rightarrow p \notin Interior(P_j)).$$

Let

$$\epsilon_{i,j} = \min_{p \in Boundary(P_i)} \{\epsilon_{p,P_j}\}. \tag{3.27}$$

Then by Lemma A.2,

$$(\forall t : 0 \leq t \leq \epsilon_{i,j}) NoIntersect(P_i, P_j, M(t)).$$

Let

$$\epsilon_P = min_{i \neq j}\{\epsilon_{i,j}\}.$$

Then $M(t) : [0, \epsilon_P]$ is a feasible motion of P (Definition 2.7), and

$$\widehat{M}(t) = \epsilon_P V t,$$

is a constant rate translational motion satisfying the theorem. ∎

3.3.2 Translational mobility is NP-hard

The proof that Problem 3.2, translational mobility, is NP-hard is a reduction of 3SAT[GJ79] to Problem 3.2. That is, it is a set of instructions that show how to construct a configuration of polygons that has a real feasible motion if and only if a given formula in 3CNF has a satisfying assignment.

Intuitively, the configuration constructed simulates a logic circuit corresponding to the 3CNF formula: a real feasible motion exists for the configuration if and only if the defining 3CNF formula is satisfiable. Thus the proof describes how to construct:

1. An **OR** gate for each of the disjunctive clauses in the formula.

2. An **AND** gate for the conjunction of the clauses.

3. A **CONSISTENCY** gate to enforce consistency of variable assignment.

4. How to combine these to get a configuration that has a real feasible motion if and only if the formula has a satisfying assignment.

An AND gate

The construct illustrated in Figure 3.6 will correspond to the conjunction of the disjunctive clauses. In the **AND** construction, polygon E can move to the right if and only if all of the vertical polygons A, B, C, and D move down. (The darker polygons will be constrained to be immobile by the construction.) Note that E can move to the right at the same rate (or less) as the slowest moving of A, B, C, and D. That is, $|V_E| \leq \min\{|V_A|, |V_B|, |V_C|, |V_D|\}$.

An OR gate

For each disjunction $(x1 + x2 + x3)$ in the formula, construct the structure illustrated in Figure 3.7. In this construction, D will be able to move down if and only if one or more of A, B, and C can move down, and $|V_D| \leq \max\{|V_A|, |V_B|, |V_C|\}$.

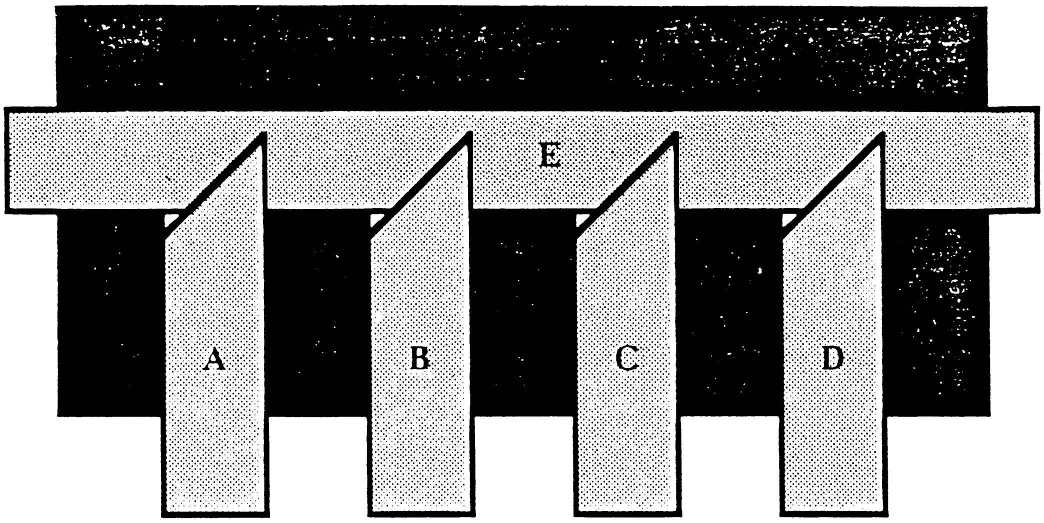

Figure 3.6: An AND Gate

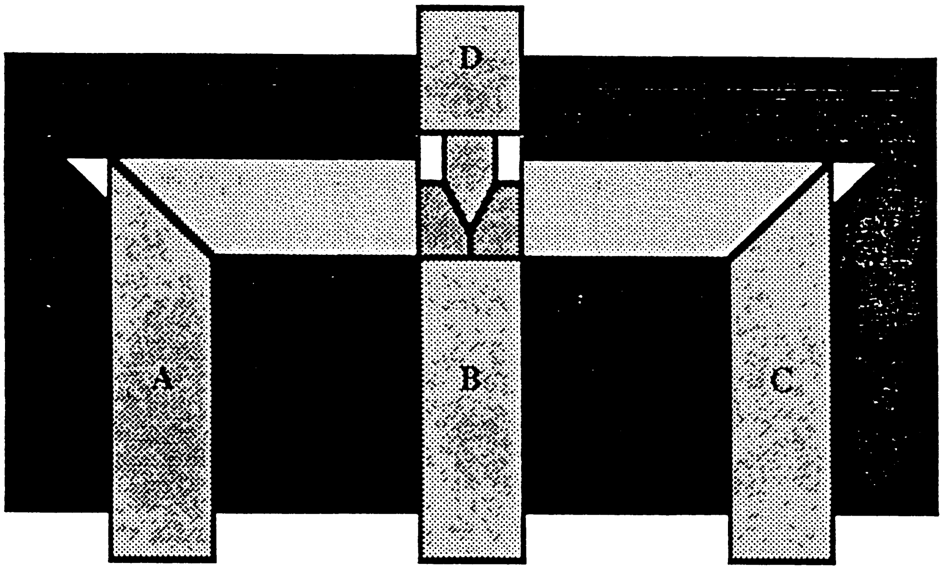

Figure 3.7: An OR gate

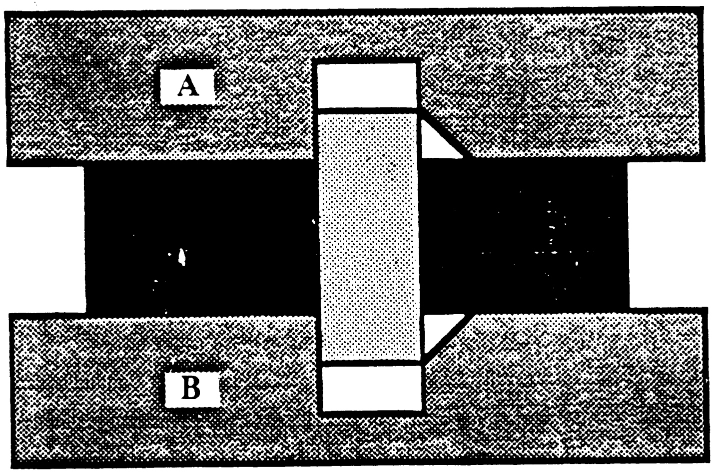

Figure 3.8: A CONSISTENCY gate

A CONSISTENCY gate

The **CONSISTENCY** gate is used to enforce the axiom $\neg(x \wedge \neg x)$. This is accomplished by the structure illustrated in Figure 3.8. In this structure only one of A and B can move to the left. Thus if A can move (x is **TRUE**) B can't ($\neg x$ is **FALSE**), and vice versa.

Combining the AND, OR, and CONSISTENCY constructions

Figure 3.9 is a schematic diagram of the final configuration. The boxes labeled $v_1, \ldots, v_k$ correspond to the **CONSISTENCY** construction for the variables of the 3CNF formula, while the boxes labeled $C_1, \ldots, C_n$ correspond to the **OR** construction for the clauses. Vertical lines correspond to polygons that will be restricted to move down or not at all. Horizontal lines correspond to polygons that will move to the left or not at all. The **AND**, **OR** and **CONSISTENCY** constructions previously described must be combined so as to allow a motion if and only if the formula is satisfiable. Crossings that are dotted correspond constructions that allow horizontal motion if and only if there is vertical motion, while crossings that are not dotted correspond to constructions that allow independent motion of the vertical and horizontal polygons.

It remains to show:

1. How to allow independent movement of vertical and horizontal lines (polygons).

2. How to allow the vertical lines corresponding to variable instances to move down if and only if the horizontal line corresponding to that variable can move left.

3. How to prevent any feasible motion unless the **AND** polygon (i.e., the polygon corresponding to the conjunction of all the clauses) can move left.

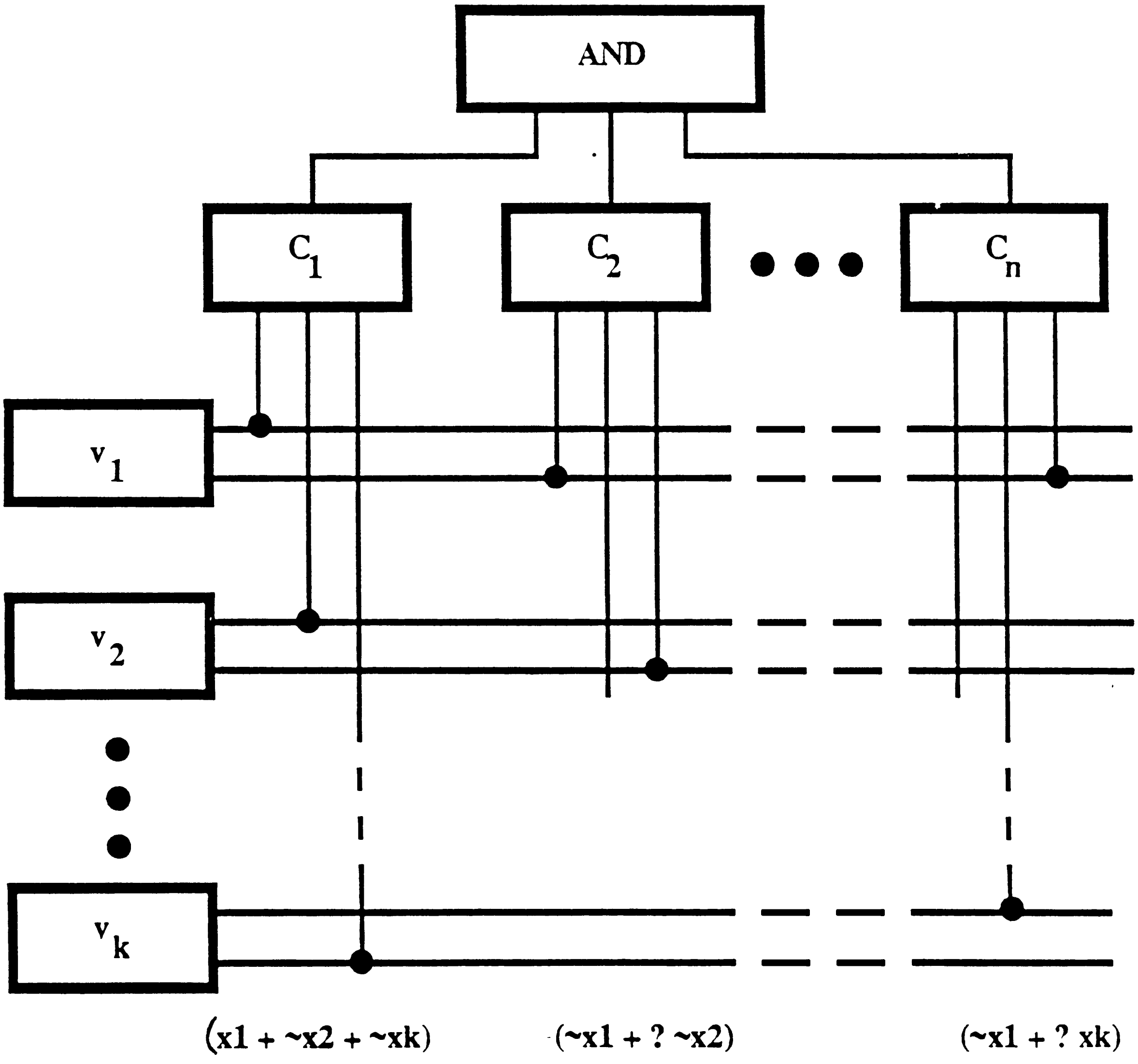

Figure 3.9: Schematic diagram of configuration

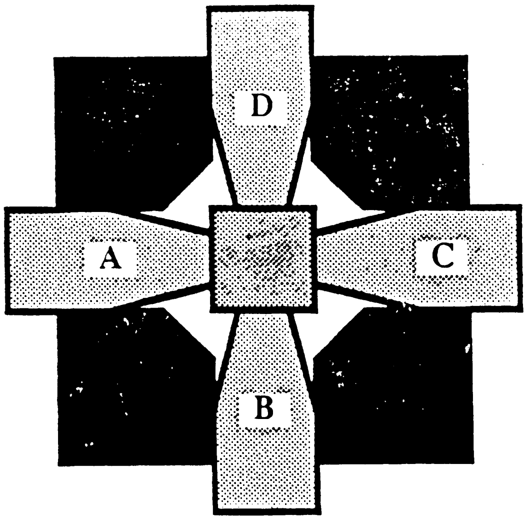

Figure 3.10: Independent vertical and horizontal translation

4. That there is a feasible motion of the configuration if and only if the 3CNF formula is satisfiable.

These goals are accomplished with the following constructs.

Independent vertical and horizontal movement

The structure illustrated in Figure 3.10 allows polygons that move vertically and horizontally to move independently. This construction is used for each independent crossing in the schematic diagram. The "independent crossings" are those crossings without a "node" in the schematic diagram Figure 3.9. (That is, those "wires" that cross in the schematic, but are not connected in the circuit.) Given velocities V_A and V_B for A and B, $|V_C| \leq |V_A|$, and $|V_D| \leq |V_B|$.

Vertical to horizontal translation

Figure 3.11 illustrates a construction that allows polygon A to move down if and only if polygon B moves to the left. Each junction with a "node" in the Figure 3.9 is replaced with a vertical to horizontal translation construct. Note that $|V_A| \leq |V_B|$.

Preventing motion

There must be no motion of any polygon in the final configuration unless the 3CNF formula has a satisfying assignment. Since the formula is true if and only if the conjunction of each disjunction is true, the configuration will be constructed so that no polygon can move unless the **AND** polygon moves. Since the **AND** polygon will be

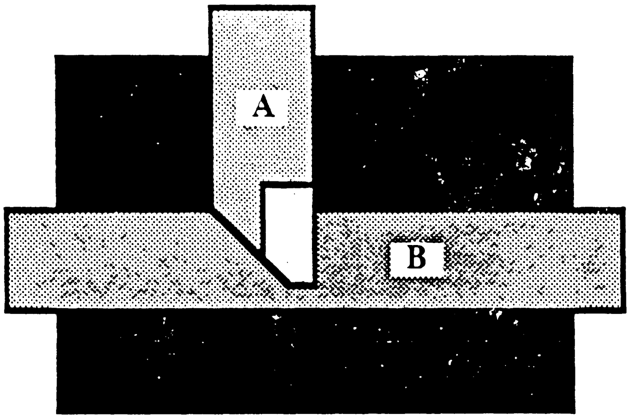

Figure 3.11: Vertical to horizontal translation

able to move if and only if the formula is satisfiable, this will ensure that no polygon can move unless the formula has a satisfying assignment.

To ensure that no polygon can move unless the **AND** polygon can move, the **CONSISTENCY** constructs are modified as shown in Figure 3.12, and a polygon Y is added to the configuration as shown in Figure 3.13. Then none of the **CONSISTENCY** polygons in Figure 3.13 will be able to move left unless Y moves down. Y will be able to move down only if **AND** can move right. Since none of the other polygons in the construction can move unless the **CONSISTENCY** constructions can, this ensures that no polygon can move unless the **AND** polygon moves. The resulting configuration is shown in Figure 3.14. In this case $|V_X| \leq V_Y$, for each polygon X in Figure 3.13.

Mobility if and only if satisfiability

This section shows that the configuration constructed in this chapter (and illustrated in Figure 3.14) has a feasible motion if and only if the defining 3CNF formula has a satisfying assignment.

Suppose that the 3CNF formula has a satisfying assignment. Then there is a consistent assignment of variables for which the formula evaluates to **TRUE**. Assuming that the Y polygon in Figure 3.13 can move down, by construction, the **CONSISTENCY** constructs allow each of the horizontal polygons corresponding to the variable assignment to move to the left. That is, suppose a variable X_i is **TRUE** in the satisfying assignment. Then the horizontal polygon corresponding to X_i = **TRUE** can move to the left.

Suppose Y moves one unit down at constant velocity of one unit per second. Then the **CONSISTENCY** constructions allow each of the "satisfied" variable polygons to move one unit to the left at this rate. By the construction of the vertical to horizontal

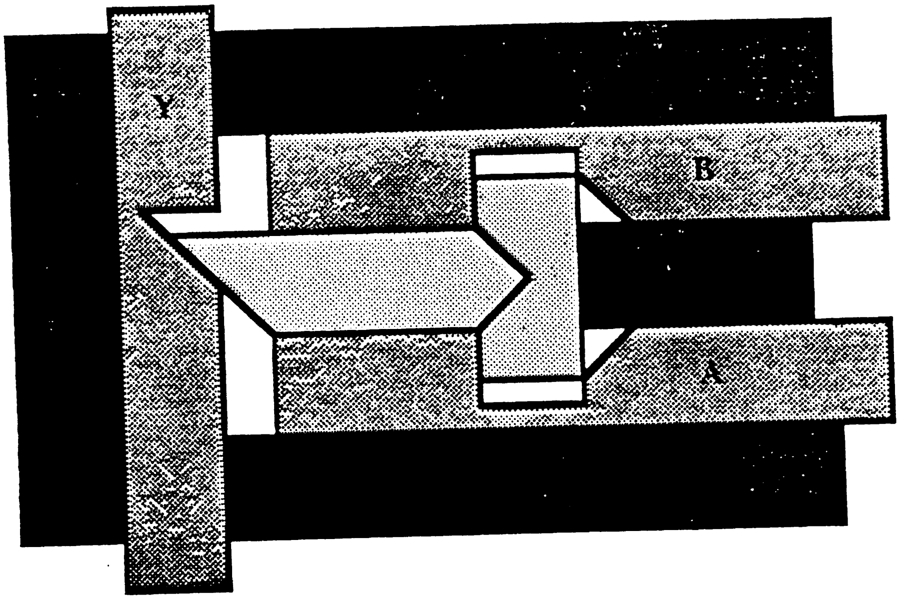

Figure 3.12: Modified **CONSISTENCY**

construct (Figure 3.11) and the independent crossing construct (Figure 3.10), each "input" polgon to an **OR** constructs can move one unit down at this rate if and only if that input corresponds to a satisfied variable. Similarly the "output" of the **OR** gates can move down one unit at this rate if and only if one of the inputs can. The **AND** construction can move to the right at this rate if and only if all output polygons of the **OR** gate do. Finally, the Y polygon in Figure 3.13 can move down at this rate if and only if the **AND** polygon can move to the right at this rate. Thus on the assumption that the formula is satisfiable, the motion described in this section is a real feasible translation of the configuration.

By construction, no motion is possible if the formula is not satisfiable. Each polygon "vertically moving" P that is constrained from moving horizontally because it is possible to find two horizontal lines passing through P that have no "gaps" between polygons: the polygons are tightly packed in the horizontal direction. A similar argument shows that a "horizontally moving" polygon P can not move vertically, and that the "fixed" polygons (the darker polygons in the construction) can not move at all.

Theorem 3.7 *Translational mobility is NP-hard.*

Proof The construction of this section. ∎

Theorem 3.8 *Mobility is NP-hard.*

Proof The construction has a real feasible motions if and only if it has a real feasible translations. ∎

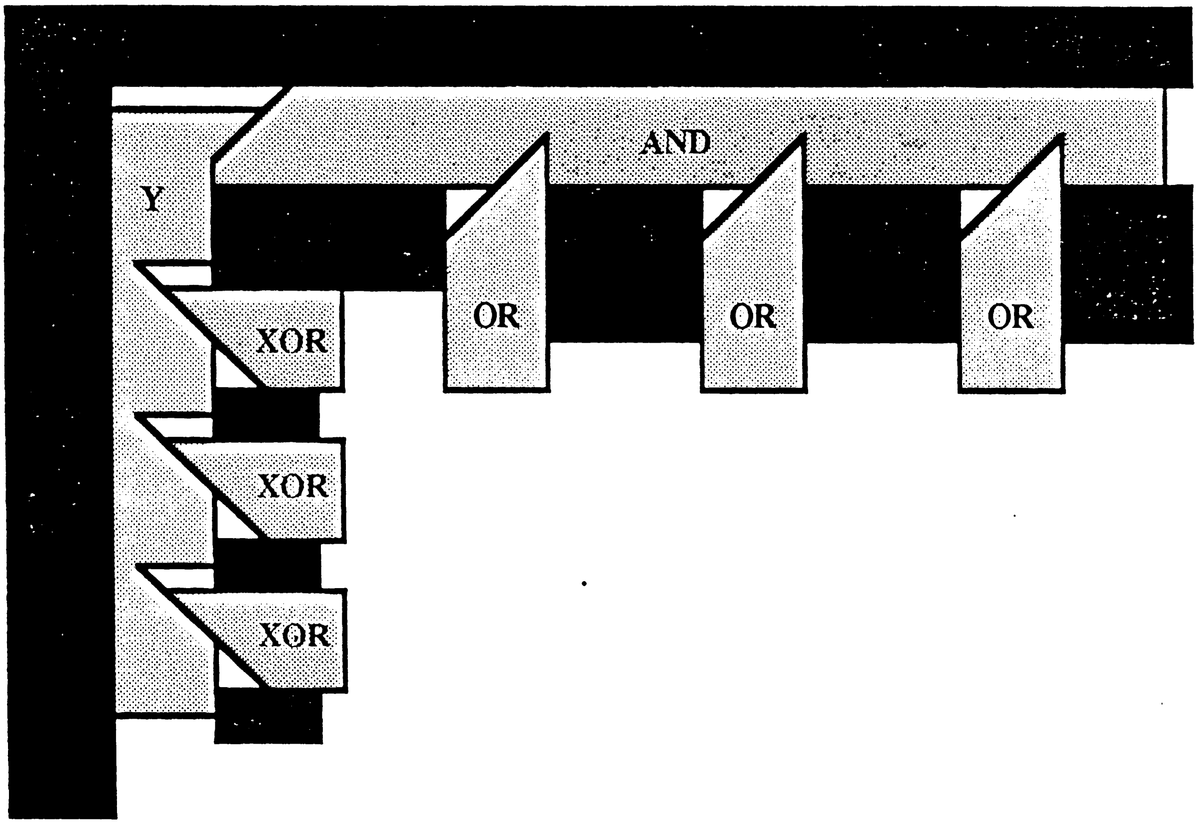

Figure 3.13: Connection of AND and CONSISTENCY gates

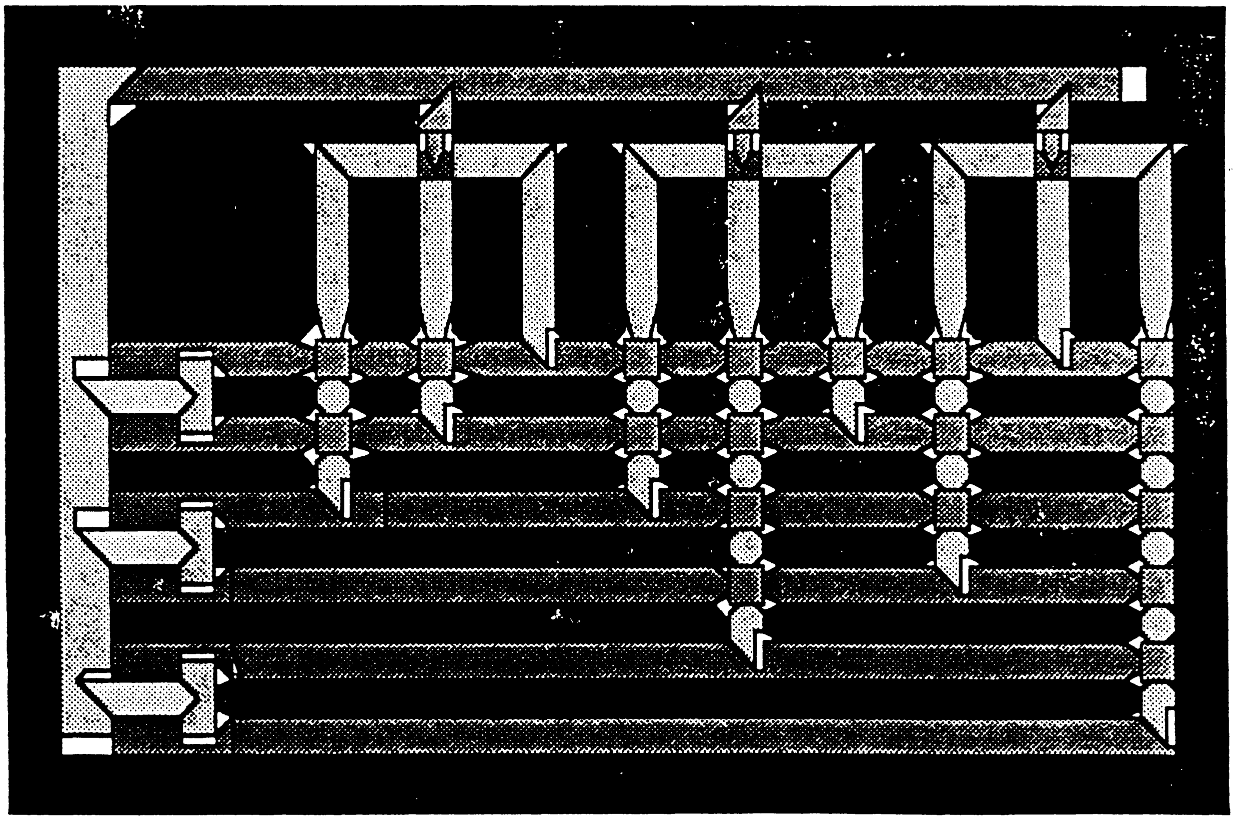

Figure 3.14: The final configuration

Theorem 3.9 *Infinitesimal mobility is NP-hard.*

Proof The construction of this chapter is infinitesimally mobile if and only if it is mobile, infinitesimal mobility is NP-hard. ∎

3.3.3 Translational mobility for configurations of convex polygons

The proof that mobility is NP-hard can be extended to a similar result for convex polygons by ensuring that the 3SAT reduction contains only convex polygons. The problem must be stated somewhat differently, however, since any configuration of convex polygons on an infinite plane has a real feasible motion.

The problem for convex polygons, the *polygonal puzzle* problem is defined thus:

Problem 3.3 (The polygonal puzzle problem) *Is there a real feasible motion for a given configuration of convex polygons enclosed in a bounded polygonal region in the plane?*

Theorem 3.10 *The polygonal puzzle problem is NP-complete.*

Proof This section contains modified constructions that employ only convex polygons. The resulting configuration contains only convex polygons, and has a real feasible motion if and only if the 3CNF formula is satisfiable. Therefore, by Theorem 3.6, the polygonal puzzle problem is NP-hard. Since the problem is contained in the translational mobility problem, it is in NP. ∎

Convex AND

In Figure 3.15, E can move right, F can move down, G, H, I, and J can move left, and K can move up if and only if all of A, B, and C move down.

Convex OR

This construct is trivially changed from the non-convex version by making its enclosing (dark) polygons convex as shown in Figure 3.16.

Convex CONSISTENCY

Figure 3.17 shows the construction for convex CONSISTENCY. The CONSISTENCY construction bears little resemblance to the non-convex CONSISTENCY construction. In Figure 3.17, exactly one of A and B can move left. No polygon can move unless E moves down in which case D and C can move down, F and G can move left, H can move either up or down, and exactly one of A and B can move left, depending on whether H moved up or down.

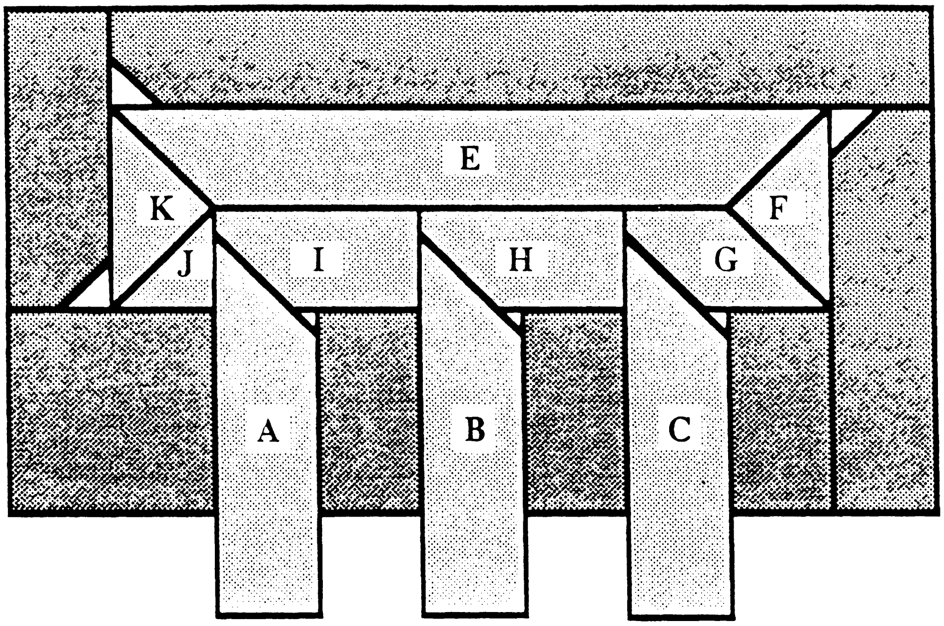

Figure 3.15: A convex AND gate

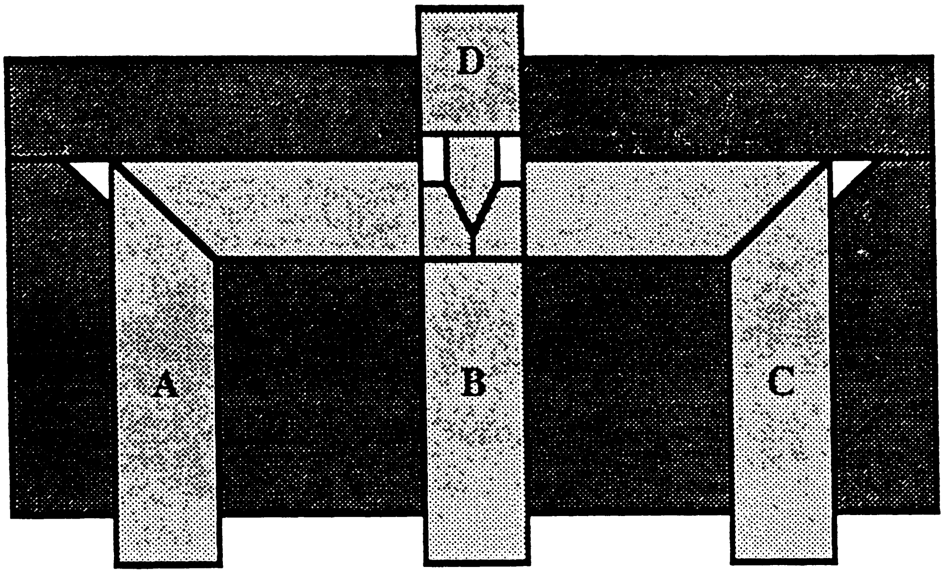

Figure 3.16: A convex OR gate

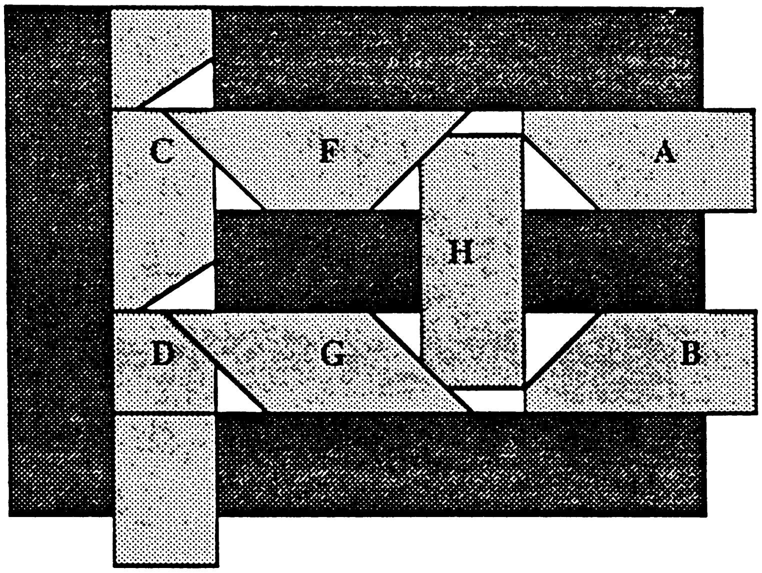

Figure 3.17: A convex CONSISTENCY gate

Convex vertical to horizontal translation

The convex version of the construction for vertical to horizontal translation is trivially changed from the non-convex version by breaking the horizontally moving polygon into convex parts. The construction for convex vertical to horizontal translation is shown in Figure 3.18.

The final convex configuration

The remaining constructions are already convex, so it remains to show how to pack them together using only convex polygons. Figure 3.19 shows how this is done. The proof that the configuration has a real feasible motion if and only if the 3CNF formula is satisfiable is similar to that of Section 3.3.2.

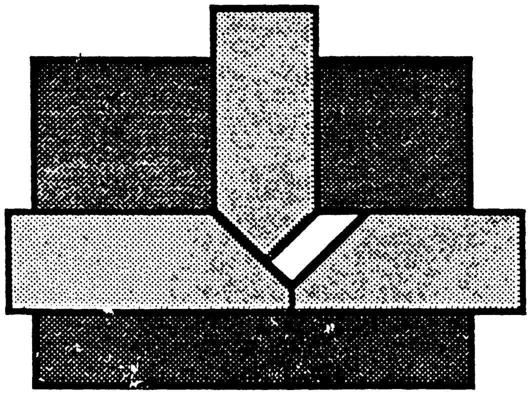

Figure 3.18: Convex vertical to horizontal translation

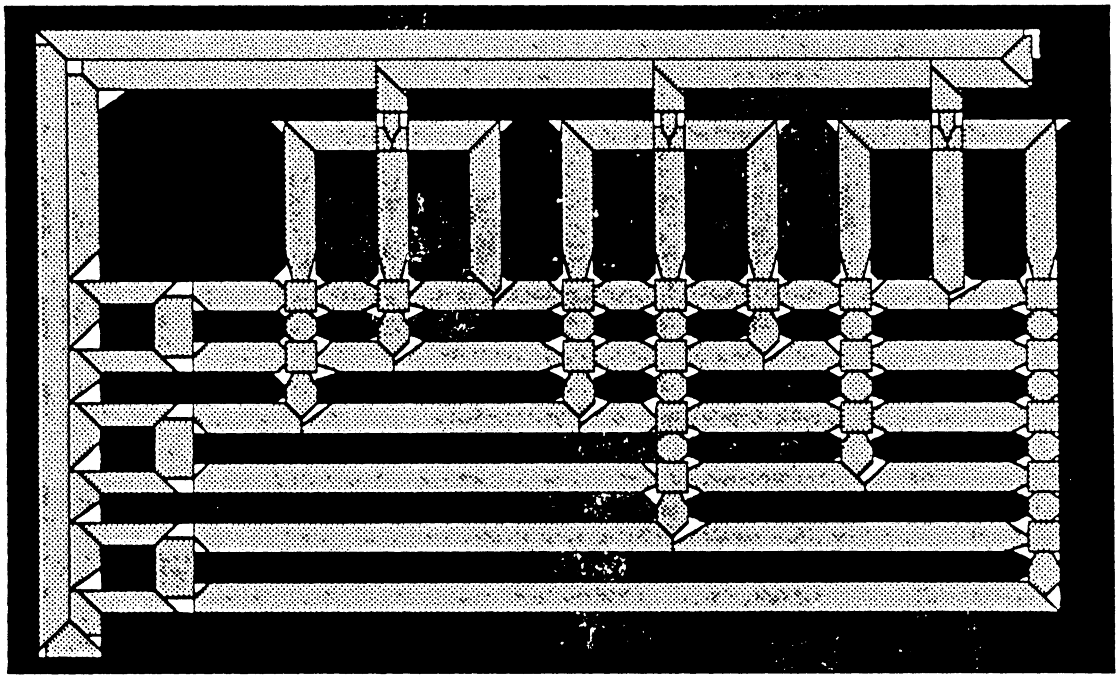

Figure 3.19: Schematic of convex polygonal configuration

Chapter 4

Stability

4.1 Introduction

The term "stability" is used in this thesis to characterize a system of rigid physical objects at a stationary static equilibrium point with respect to some inertial reference frame. This differs from the usual use of the term in engineering in that a configuration that is "stable" (as the term is used in this thesis) may be at either a stable or unstable static equilibrium point. (A broom balancing on it's handle is at an unstable equilibrium point; a broom lying on the floor is at a stable equilibrium point.)

The notion of a stable configuration of objects is clear: a set of oriented physical objects is stable if the relative position of the objects remains unchanged over time. In the context of robotic assembly, we assume that objects are initially fixed in space, and say that they are stable if the force of gravity cannot cause the position of any object to change.

For polygonal configurations modeling physical objects, the notion of stability is similarly clear. Suppose we were to "construct" a configuration of polygons by "pinning" each polygon in its position. Intuitively, we would consider the configuration stable if we could then simultaneously remove the "pins," and the configuration would be guaranteed to remain in that position.

4.2 Modeling statics

This section describes two characterizations of stable configurations of polygons: those configurations for which there is a set of force equations that satisfy the *force conditions* for stability, and those for which there is no feasible *positive work* infinitesimal motion. Both methods derive from Newton's model of mechanics and are closely related, as will be seen in Section 4.5. Since any introductory text in physics [HR62] or mechanical engineering [Sh80] contains a development of these methods, I shall simply state the laws and conditions.

4.2.1 Mechanics

1. The momentum of a body is constant if there are no net external forces acting on it.

2. $F = ma$, where F is the resultant external force, m is the mass, and a is the acceleration of the body. The acceleration a can be decomposed into *translational* and *rotational* acceleration. Translational acceleration is the rate of change in velocity of the center of mass to change, while rotational acceleration is the rate of change in the rotational velocity about the center of mass.

3. If a body A exerts a force F_{AB} on a body B, then B exerts a force F_{BA} on A, and F_{BA} has the same magnitude and opposite direction as F_{AB} (i.e., $F_{AB} = -F_{BA}$).

4.2.2 Force

Definition 4.1 In a configuration of polygons, the force of gravity is *active.* Other forces are *contact forces,* which occur only at points $p \in Contact(P_i, P_j)$. Contact forces are *reactive.*

Notation 4.1 *weight(P_i) denotes the force of gravity on P_i. Choosing as a gravitational constant $g = 1$, and $(0, -1)$ as the direction of gravity, $weight(P_i) = (0, -mass(P_i))$, and is located at $cm(P_i)$, the point at which the mass of P_i is located.*

Definition 4.2 For a given configuration of polygons P, define a *force assignment* $F(P)$ to be an assignment of a force vector $f_n = (f_x, f_y)$ to each node in the configuration. Thus if

$$P = (P_1, \ldots, P_n) \tag{4.1}$$

and each

$$P_i = (n_{i,1}, \ldots, n_{i,k_i}) \tag{4.2}$$

then

$$F(P_i) = (f_{n_{i,1}}, \ldots, f_{n_{i,k_i}}) \tag{4.3}$$

and

$$F(P) = (F(P_1), \ldots, F(P_n)). \tag{4.4}$$

Note that there is one force vector f_n in $F(P)$ for each node n in P.

Definition 4.3 A vector field is *conservative* if it is the gradient of a scalar valued function of a vector. A force F is *conservative* if it is the gradient of of a time independent potential.

The force of gravity on a body is conservative, since it is modeled as a constant, while the force of friction is non-conservative, since it depends on velocity.

4.2.3 Configurations without friction

For configurations assumed to have frictionless surfaces, the force assignment must satisfy the constraint that reactive forces be normal to surfaces of contact. In the case of frictionless polygons, this means that the reactive force f_n at a node n of P_i must satisfy the following condition:

Condition 4.1 *Suppose* $n = e_1 \cap e_2$. *Then*

$$f_n = c_1 N(e_1) + c_2(N(e_2)), \tag{4.5}$$

for $c_1, c_2 \geq 0$.

4.2.4 Configurations with friction

If the polygons are modeled as having friction, then for each distinct P_i and P_j in the configuration, μ_{ij}, the *coefficient of friction* defines the maximum ratio between the normal and tangential components of a force at any point of contact between P_i and P_j. Thus, if there is friction between polygons, the reactive forces must satisfy the following condition:

Condition 4.2 *Suppose* f_p *is a reactive force at* $p \in Contact(P_i, P_j)$, *and that* μ_{ij} *is the coefficient of friction for* p *between* P_i *and* P_j. *Suppose* $p = e_1 \cap e_2$, *for* e_1, e_2, *edges of* P_j.

Then

$$f_p = f_{e_1} + f_{e_2} \tag{4.6}$$

subject to the following constraints:

$$f_{e_i} = T_{e_i} + N_{e_i} \tag{4.7}$$

$$T_{e_i} \leq \mu_{ij} N_{e_i}, \tag{4.8}$$

where T_{e_i} *is tangent to* e_i, *and* N_{e_i} *is normal to* e_i.

4.2.5 Stability conditions

To say that a configuration is stable under a force assignment is to say that it is not accelerating. Setting $a = 0$, (i.e., positing that there is no rotational or translational acceleration) in the second law of mechanics implies $F = 0$, i.e., that the resultant of all forces on a body is the zero vector. Together with the third law of mechanics, this yields conditions for stability.

Theorem 4.1 (Stable force conditions) *For rigid bodies, a configuration is stable (i.e., at a static equilibrium point) only if there is a set of forces that satisfy the following conditions:*

1. *The sum of all forces on each body is zero.*

2. *The moment induced by all forces on a body about any point is zero.*
3. *If a body A exerts a force F_{AB} on a body B, then B exerts a force F_{BA} on A, and F_{BA} has the same magnitude and opposite direction as F_{AB} (i.e., $F_{AB} = -F_{BA}$).*

Additional conditions for the stability of configurations of polygons

1. All reactive forces are located at nodes of polygons. This assumption is justified by Lemma 4.1.
2. All reactive forces are directed to the interior of the body, since the model is not intended to include bodies that are "glued" together. This means that each reactive force f_n, for n a node of P_i, must satisfy $\mathrm{DirExt}(n, P_i, -f_n)$. (Definition 3.1.) That is, the force f_n must be directed towards the interior of P_i in the neighborhood of n.

4.2.6 Location of reactive forces

If an algorithm is to test for stability in polynomial space, the space required to represent the forces must be polynomial in the size of the input (i.e., in the size of P). Therefore, assume that all reactive force between a pair of polygons P_i and P_j in contact occurs at points in $Contact(P_i, P_j)$ that coincide with a node of either P_i or P_j. This assumption is justified by the following lemma.

Lemma 4.1 *Suppose that, for a configuration P of polygons with mass, there is a set of forces that satisfy the stability conditions (or more generally, Newton's laws of mechanics.) Then there is a satisfying set with all reactive forces occurring at points of node contact.*

Proof The proof shows that any force that occurs on an edge of contact (not at a node) can be replaced by two forces, one at each endpoint of the edge, without changing the resultant of the forces on the polygon, and that forces can be replaced in pairs to satisfy the third law of mechanics. Figure 4.1 illustrates the proof.

Suppose a force f_p is acting at point p on an edge of contact $e = (n_1, n_2)$. Let $l_1 = |p - n_1|$, and $l_2 = |p - n_2|$. Then f_p can be replaced with the pair

$$f_1 = \frac{l_2}{l_1 + l_2} f_p \tag{4.9}$$

$$f_2 = \frac{l_1}{l_1 + l_2} f_p. \tag{4.10}$$

The resultant of f_1 and f_2 is f_p, which follows from the following argument. Clearly $f_p = f_1 + f_2$, since

$$f_1 + f_2 = \frac{l_2}{l_1 + l_2} f_p + \frac{l_1}{l_1 + l_2} f_p \tag{4.11}$$

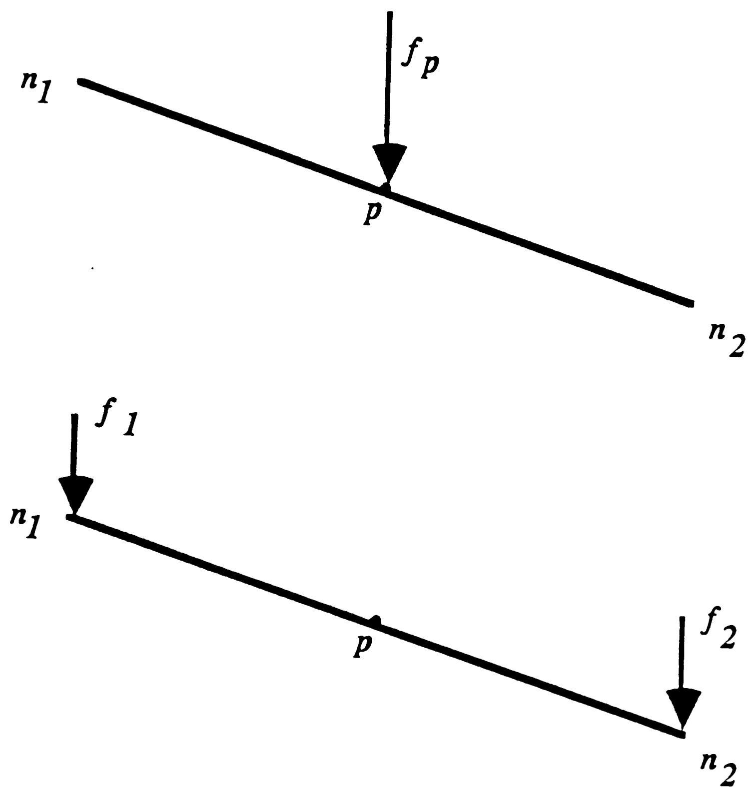

Figure 4.1: Replacement of a force acting on an edge.

$$= (\frac{l_2}{l_1 + l_2} + \frac{l_1}{l_1 + l_2}) f_p \quad (4.12)$$

$$= f_p. \quad (4.13)$$

Thus it remains to show that f_1 and f_2 do not impose a torque about p. It suffices to show that the torque about any point is zero. Consider the torque about the point n_1. The torque imposed by the pair f_1 and f_2 about p is given by

$$f_1 \times \frac{e}{l_1} + f_2 \times \frac{-e}{l_2} \quad (4.14)$$

$$= \frac{l_2}{l_1 + l_2} f_p \times \frac{e}{l_1} + \frac{l_1}{l_1 + l_2} f_p \times \frac{-e}{l_2} \quad (4.15)$$

$$= 0. \quad (4.16)$$

Since f_1 and f_2 are parallel to f_p, they satisfy Condition 4.2 (or Condition 4.1) if and only if f_p does.

If the original set of forces satisfied the third law of mechanics, both forces in a reactive pair can be replaced in this manner, and the resulting force set also satisfies the third law. ∎

4.2.7 Conditions for instability

This section describes conditions that any infinitesimally unstable configuration satisfies, and that no unconditionally stable configuration satisfies. These conditions will be used to derive algorithms for determining whether a given configuration P is stable.

Definition 4.4 The *work* done by a set of forces F in moving the polygons of P from $M(0)$ to $M(t)$ is given by

$$W(t) = \sum_p \int_0^t f_p(t) \cdot V_p(t)\, dt, \quad (4.17)$$

where p ranges over the set of points the forces of F are acting on, $f_p(t)$ is the force on p at t, and $V_p(t)$ is the velocity of p at t [HR62].

Theorem 4.2 (Virtual work) *A body or system of bodies is stable if and only if* $W'(t) \leq 0$ *for any feasible velocity assignment* V. [Sh80].

For a set of polygons P, the work function can be decomposed into three components:

1. The work done by the contact forces, i.e., the work done by friction.
2. The work resulting in translational acceleration of the polygon.
3. The work resulting in rotational (i.e., angular) acceleration of the polygon.

Thus the work function is given by

$$W(t) = W_{contact}(t) + W_{angular}(t) + W_{translational}(t), \tag{4.18}$$

where

$$W_{contact}(t) = \sum_{1\le i\le n} \sum_{p\in Contact(P_i,P_j)} \int_0^t \frac{1}{2} f_p \cdot V_p{}^j(t)\, dt \tag{4.19}$$

$$W_{angular}(t) = \sum_{1\le i\le n} \int_0^t MIcm(i)\theta_i{}'(t)\, dt \tag{4.20}$$

$$W_{translational}(t) = \sum_{1\le i\le n} \int_0^t weight(i) \cdot V_{cm(i)}(t)\, dt. \tag{4.21}$$

The correctness of this decomposition is shown by the following lemma.

Lemma 4.2 *For a configuration of polygons P, the work function given in Equation 4.17 can be decomposed as in Equation 4.18.*

Proof In a configuration of polygons, all forces are one of two types:

1. Contact force occurring at a node p of some polygon P_i.
2. The force of gravity on the mass of a polygon P_i.

Thus the work for a configuration of polygons is given by

$$W(t) = \sum_{1\le i\le n} \sum_{p\in P_i} \int_0^t f_p(t) \cdot V_p(t)\, dt, \tag{4.22}$$

$$+ \sum_{1\le i\le n} \int_0^t weight(i) \cdot V_{cm(i)}(t)\, dt. \tag{4.23}$$

It remains to show that

$$\sum_{1\le i\le n} \sum_{p\in nodes(P_i)} \int_0^t f_p(t) \cdot V_p(t)\, dt \tag{4.24}$$

$$= \sum_{1\le i,j\le n} \sum_{p\in Contact(P_i,P_j)} \int_0^t \frac{1}{2} f_p \cdot V_p{}^j(t)\, dt \tag{4.25}$$

$$+ \sum_{1\le i\le n} \int_0^t MIcm(i)\theta_i{}'(t)\, dt. \tag{4.26}$$

Note that $V_p(t)$ in Equation 4.22 is an absolute velocity, while $V_p{}^j(t)$ in Equation 4.19 is relative to the velocity of P_j.

The force f_n occurring at a node n can be decomposed into two forces: the tangential force T_{f_n}, and the normal force N_{f_n}, such that

$$f_n = T_{f_n} + N_{f_n}. \tag{4.27}$$

Consider a pair of reactive forces f_{n_i} and f_{n_j} acting at a point $p = n_i = n_j$, for nodes of P_i and P_j, respectively. By the third law of mechanics, $f_{n_i} = -f_{n_j}$. Let V_{n_i} and V_{n_j} denote the absolute velocities of n_i and n_j, respectively. The normal forces of reactive pairs cancel out in the work function since

$$\begin{aligned} N_{f_{n_i}} &= -N_{f_{n_j}}, & (4.28)\\ V_{n_i} &= V_{n_j}, & (4.29) \end{aligned}$$

and thus

$$V_{n_i} \cdot N_{f_{n_i}} + V_{n_j} \cdot N_{f_{n_j}} = 0. \tag{4.30}$$

Consider the work done by the tangential components of the reactive pair.

$$T_{f_{n_i}} = -T_{f_{n_j}}, \tag{4.31}$$

and thus the infinitesimal work done by the pair is

$$\begin{aligned} & V_{n_i} \cdot T_{f_{n_i}} + V_{n_j} \cdot T_{f_{n_j}} & (4.32)\\ =\ & (V_{n_i} - V_{n_j}) \cdot T_{f_{n_i}} & (4.33)\\ =\ & V_{n_i}{}^{j} \cdot T_{f_{n_i}} & (4.34)\\ =\ & V_{n_j}{}^{i} \cdot T_{f_{n_j}} & (4.35)\\ =\ & \frac{1}{2} V_{n_i}{}^{j} \cdot T_{f_{n_i}} + \frac{1}{2} V_{n_j}{}^{i} \cdot T_{f_{n_j}} & (4.36) \end{aligned}$$

By the second law of mechanics, the resultant of the forces on each polygon must result in acceleration. Decomposing the resultant into its translational and angular components results in Equation 4.20 and Equation 4.21. ∎

Thus

$$W'(t) = W'_{contact}(t) + W'_{angular}(t) + W'_{translational}(t), \tag{4.37}$$

where

$$\begin{aligned} W'_{contact}(t) &= \sum_{1 \le i,j \le n} \sum_{p \in nodes(Contact(P_i,P_j))} \frac{1}{2} f_n \cdot V_n{}^{j}(t) & (4.38)\\ W'_{angular}(t) &= \sum_{1 \le i \le n} MIcm(i)\theta_i'(t) & (4.39)\\ W'_{translational}(t) &= \sum_{1 \le i \le n} weight(i) \cdot V_{cm(i)}(t). & (4.40) \end{aligned}$$

If a configuration of polygons P is unstable, then $W'(0) > 0$ for some feasible velocity V.

4.3 Stability problems

This section defines the problems considered in this chapter. Each problem concerns determining the behavior of a given configuration of polygons with mass P. Each of these problems applies either in the presence or absence of friction.

1. Stability: Is the configuration guaranteed to be stable?
2. Potential instability: Is there a chance that the configuration is unstable?
3. Infinitesimal instability: Is there an infinitesimal motion for which the configuration is unstable?
4. Infinitesimal stability: Is there no infinitesimal motion for which the configuration is unstable? This is the complement of the infinitesimal instability problem, and a conservative approximation to the stability problem.
5. Instability: Is the configuration guaranteed to be unstable?
6. Potential stability: Is there any possibility that the configuration is stable? (The complement of the instability problem.)

4.4 Stability problems without friction

This section investigates the computational complexity of these stability problems using the frictionless model.

4.4.1 Frictionless infinitesimal instability is in NP

The method of virtual work states that a configuration is stable if there is no feasible velocity that is a positive work direction for any feasible force assignment $F(P)$.

Theorem 4.3 *Frictionless infinitesimal instability is in NP.*

Proof By Theorem 4.2, a configuration is unstable if and only if the infinitesimal work function for P, $W'(t)$, is positive. Equation 4.37 describes $W'(t)$ for configurations of polygons. In the frictionless model, all contact forces are normal to surfaces of polygons: no work is done by any contact force. This follows from the fact that for any $n_i \in Contact(P_i, P_j)$, $f_n(t) \cdot V_{n_i}{}^j(t) = 0$, since for any n, either:

1. $V_{n_i}{}^j(t)$ is tangent to the surface, and hence normal to $f_n(t)$,
2. $V_{n_i}{}^j(t) = 0$, or
3. $V_{n_i}{}^j(t) \cdot N(e) > 0$, (i.e., $V_n(t)$ is not tangent to an edge), in which case $f_n = 0$, since no force can occur where the velocity would cause contact to be broken.

Therefore, the work function for a configuration of frictionless polygons is given by

$$W(t) = W_{angular}(t) + W_{translational}(t) \tag{4.41}$$

and thus

$$W'(t) = \sum_{1 \leq i \leq n} (MIcm(i)\theta_i'(t) + weight(i) \cdot V_n(t)). \tag{4.42}$$

Thus a configuration P is infinitesimally unstable if and only if there is a feasible infinitesimal motion for which Equation 4.42 is positive. The results of Section 3.2.1 show that infinitesimal mobility is in NP. Therefore, by Lemma A.4, infinitesimal instability is in NP if Equation 4.42 can be formulated as a linear objective function with with rational coefficients whose size is polynomial in the size of P.

If $V_{cm(i)}(t) = (x_{cm(i)}(t), y_{cm(i)}(t))$, then

$$\sum_{1 \leq i \leq n} weight(i) \cdot V_{cm(i)}(t) + MIcm(i)\theta_i'(t) \tag{4.43}$$

$$= \sum_{1 \leq i \leq n} -mass(P_i) y_{cm(i)}(t) + MIcm(i)\theta_i'(t) \tag{4.44}$$

which is a linear objective function with rational coefficients $mass(P_i)$ and $MIcm(i)$ which are in fact part of the input P. ∎

Frictionless infinitesimal stability is the problem of determining that there are no feasible positive work infinitesimal motions. This is a conservative approximation to the general stability problem.

Corollary 4.1 *Frictionless infinitesimal stability is in CO-NP.*

Proof Frictionless infinitesimal stability is the complement of frictionless infinitesimal instability. ∎

4.4.2 Potential instability is NP-hard

The proof that infinitesimal instability is NP-hard is similar to the proof that Problem 3.1 is NP-hard. That is, infinitesimal instability is shown to be NP-hard by showing how to construct a configuration of polygons that is infinitesimally unstable if and only if a given formula in 3CNF is satisfiable. In fact, the configuration constructed is identical to that constructed in Chapter 3, except for the addition of mass to the polygons.

The mass is assigned to polygons so that if any infinitesimal motion is feasible, a positive work infinitesimal motion is feasible, and thus by Theorem 4.2, the configuration is infinitesimally unstable.

Theorem 4.4 *Frictionless infinitesimal instability is NP-hard.*

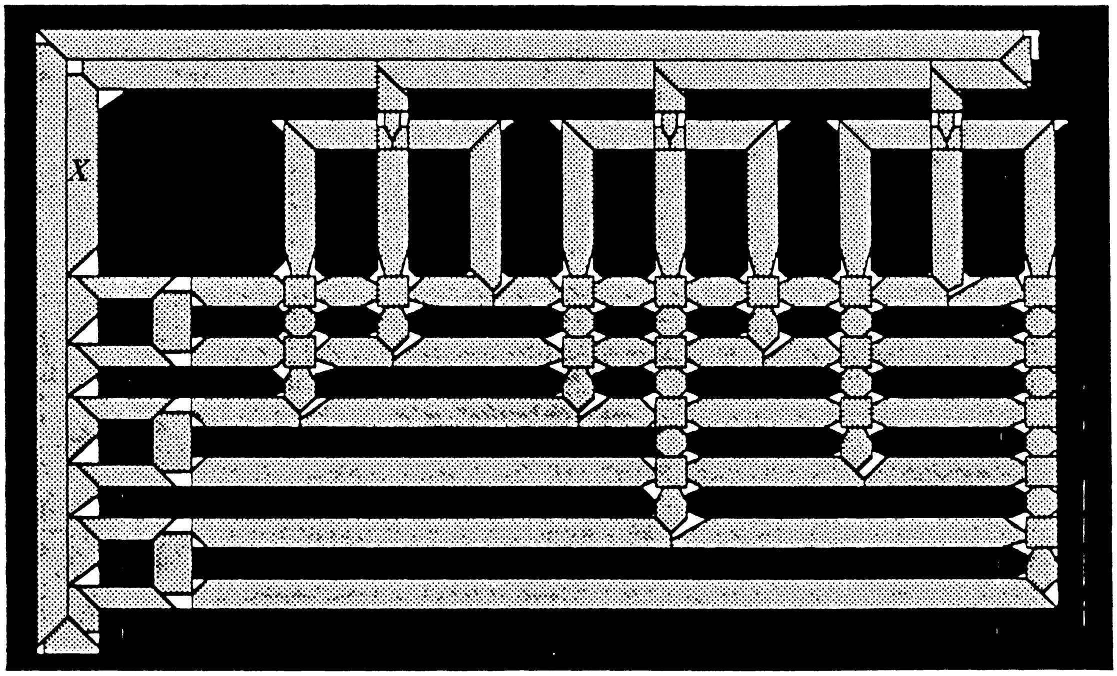

Figure 4.2: A schematic diagram of the *instability* configuration

Proof The schematic of the final configuration is shown in Figure 4.2. All polygons except that labeled X in Figure 4.2 are assigned a mass of 1. Polygon X is assigned a mass equal to twice the sum of the mass of the other polygons.

The proof of Theorem 3.7 showed that there is a feasible motion for the configuration if and only if the 3CNF formula is satisfiable, and that no motion is possible unless X can move down. Let $-y$ be the direction that X can move. Since the mass of X is greater than the sum of the masses of the other polygons in the configuration, and no other polygon can move in the $-y$ direction at a greater velocity that X, it follows that there is a positive work infinitesimal motion if and only if the 3CNF formula is satisfied. ∎

Theorem 4.5 *Potential instability is NP-complete.*

Proof Theorem 4.4.2 and Theorem 4.3. ∎

Corollary 4.2 *Frictionless infinitesimal stability is CO-NP-complete.*

Proof Frictionless infinitesimal stability is the complement of the frictionless infinitesimal instability problem. ∎

Corollary 4.3 *Frictionless stability is CO-NP-hard.*

Proof The configuration in Figure 4.2 is guaranteed to stable if and only if the 3CNF formula is not satisfiable. ∎

4.4.3 Frictionless potential stability is in P

This section shows that potential stability is in P, i.e., that there is a deterministic polynomial algorithm to determine whether there is a set of forces that satisfy Theorem 4.1 and Condition 4.1. Recall that if it is possible for a given configuration P to be stable, then there must be a set of forces F that satisfy the constraints of Section 4.2.5. Each of these constraints can be formulated as a linear inequality with rational coefficients. The result is a linear program whose size is polynomial in the size of P. Thus potential stability is in P.

Theorem 4.6 *Potential stability is in P.*

Proof For a configuration P with frictionless surfaces to be stable, the following constraints must be satisfied:

1. The sum of external forces on each polygon must be zero. For each polygon P_i,

$$\sum_{1\leq j\leq k_i} f_{n_j} = -weight(P_i). \tag{4.45}$$

This can be formulated as a pair of linear inequalities:

$$\begin{aligned} f_{n_{i,1}} + \cdots + f_{n_{i,k_i}} &\leq weight(P_i) \\ f_{n_{i,1}} + \cdots + f_{n_{i,k_i}} &\leq -weight(P_i) \end{aligned}$$

Since $weight(P_i)$ is rational (and all other coefficients are 1), all coefficients are rational, and their total size is polynomial in the size of P.

2. The moment induced about any point p by all forces on each polygon must be zero. Thus, choosing $p = (0,0)$,

$$\sum_{1 \leq j \leq k_i} (f_{n_j} \times n_j) = -(weight(P_i) \times cm(P_i)) \tag{4.46}$$

Each constraint $f_n \times n$ can be formulated as a linear equality, since if $f = (f_x, f_y)$, and $n = (n_x, n_y)$,

$$\begin{aligned} f \times n &= \begin{vmatrix} i & j & k \\ f_x & f_y & 0 \\ n_x & n_y & 0 \end{vmatrix}, \\ &= k(f_x n_y - f_y n_x). \end{aligned}$$

3. $F_{AB} = -F_{BA}$. Thus if n_1 is a node of P_i and n_2 is a node of P_j, then $f_{n_1} = -f_{n_2}$. This is a linear equality with rational coefficients.

4. All reactive forces are normal to the surface of contact. Thus

$$f_{n_i} = c_1 N(e_i) + c_2 N(e_{i+1}) \tag{4.47}$$

for all n_i in P_j, for $c_1, c_2 \leq 0$. This can be formulated as a set of linear inequalities:

$$\begin{aligned} -f_{n_i} &= f_1 + f_2 && (4.48) \\ f_1 \cdot N(e_i) &= 0 && (4.49) \\ f_2 \cdot N(e_{i+1}) &= 0 && (4.50) \end{aligned}$$

Thus the potential stability problem can be described as a linear program. Let LP denote the linear program constructed in this section. It is straightforward to verify that $Size(LP) \overset{P}{\preceq} Size(P)$. Thus potential stability is in P. ∎

Corollary 4.4 *Frictionless instability is in P.*

Proof Frictionless instability is the complement of frictionless potential stability. ∎

4.5 Stability problems with friction

This section investigates the computational complexity of stability problems for configurations of polygons in the presence of friction.

4.5.1 Infinitesimal instability with friction

The class of configurations of polygons with friction includes the frictionless configurations, since a configuration with $\mu_{ij} = 0$, for all pairs of polygons in P, is a frictionless configuration.

Theorem 4.7 *Infinitesimal instability for configurations of polygons with friction is NP-hard.*

Proof The proof is a reduction of frictionless infinitesimal instability to infinitesimal instability with friction. ∎

Corollary 4.5 *Potential instability with friction is NP-hard.*

Proof The proof is a reduction of frictionless stability to stability with friction. ∎

4.5.2 Infinitesimal instability is in NP for configurations of polygons with friction

The mechanics of a configuration of polygons with friction is modeled with a system of equations that ensures that the laws of mechanics are satisfied. This is a generalization of the methods used to model the frictionless problem: in fact it is a combination of force conditions for stability and the method of virtual work for frictionless configurations.

Suppose a configuration of polygons P is initially stationary, and is unstable. Then there is a set of reactive forces $F(P)$ such that

1. $F(P_i) = ma$, where
$$m = (mass(1), \ldots, mass(n)),$$
and
$$a = (A_{cm(1)}, \ldots, A_{cm(i)}(t)).$$

2. $F(P)$ must satisfy the first and third laws of motion.

3. For each polygon P_i in P, the resultant torque imposed by the force assignment $F(P_i)$ must equal the angular acceleration, i.e.,
$$\sum_{n \in P_i} f_n \times n = \theta_i' MIcm(i)k, \tag{4.51}$$
where k is the unit vector orthogonal to the plane.

4. The acceleration a must be consistent with the feasibility constraints of infinitesimal motion. That is, there must be a feasible motion M with acceleration a.

5. $W' > 0$, since Theorem 4.2 states that the infinitesimal work $W' > 0$ if the configuration is unstable.

The virtual work formulation for the model including friction is more complicated that that for the frictionless model, because the net infinitesimal work done by a force and velocity assignment depends on the force of friction. Unlike the frictionless case, a reactive force f_{n_i} at a point in $Contact(P_i, P_j)$ is not necessarily orthogonal to the direction of $V_n{}^j$, and thus reactive force must be considered in the virtual work formulation.

An NP algorithm can determine that a configuration is unstable by guessing a set of forces and an infinitesimal motion, and verifying that they satisfy the constraints for motion and positive work.

The work function for a configuration of polygons P with friction is given by

$$W(t) = contact(t) + W_{angular}(t) + W_{translational}(t). \tag{4.52}$$

The formula for $W'(t)$ is

$$\begin{aligned} W'(t) &= \sum_{n \in P} f_n(t) \cdot V_n(t) && (4.53) \\ &+ \sum_{1 \le i \le n} (MIcm(i)\theta_i'(t) weight(i) \cdot V_{cm(i)}(t)(t)). && (4.54) \end{aligned}$$

Feasible force and motion assignments for configurations of polygons with friction

The feasibility constraints for $f_n(t)$ depend on $V_n(t)$, the velocity at n.

Suppose n is a node of P_i, that $n \in Contact(P_i, P_j)$, and that $n = e_1 \cap e_2$, for e_1 and e_2, edges of some polygon P_j. Let V_i^j, the relative velocity of n with respect to P_j.

There are three cases:

1. If the velocity of n_i is away from the surface of P_j, then $f_{n_i} = 0$, since the only forces other than gravity are contact forces. Thus

$$(V_i^j \cdot N(e_1) > 0 \wedge V_i^j \cdot N(e_2) > 0) \Rightarrow f_{n_i} = 0. \tag{4.55}$$

Since $f_{n_i} = 0$, the infinitesimal work $f_{n_i} \cdot V_i^j = 0$.

2. If $V_i^j = 0$, then f_n must satisfy Condition 4.2. Thus if $V_i^j = 0$, then

$$f_n = f_{e_1} + f_{e_2} \tag{4.56}$$

subject to the constraints that

$$\begin{aligned} f_{e_i} &= T_{e_i} + N_{e_i} \\ T_{e_i} &\le \mu_{ij} N_{e_i}, \end{aligned}$$

where T_{e_i} is tangent to e_i, and N_{e_i} is normal to e_i. Since there is no velocity at n, $f_n \cdot V_i^j = 0$, i.e., there is no infinitesimal work done by f_n at n.

3. If $V_i^j \neq 0$ is tangent to some edge e_j of P_j, then this motion is opposed by friction. To be consistent with motion, the frictional force must oppose V_i^j, (i.e., $V_i^j \cdot f_n \leq 0$), and the ratio of the magnitudes of its tangential and normal components must be μ_{ij}.

   In effect, this will mean that
   $$f_n = T + N \tag{4.57}$$
   where $T \cdot N = 0$, $N \cdot e_j = 0$, and $|T| = \mu_{ij}|N|$. The infinitesimal work done by f_n is then $f_n \cdot V_i^j$.

By Theorem 4.2, P is infinitesimally unstable if there is a $V = M'(0)$ that is consistent with the mechanical model, and is such that $W'(0) < 0$. Thus, a force assignment must satisfy Newton's second and third laws, in particular. Section 4.4.1 showed how to constrain $F(P)$ so that the third law is satisfied.

To constrain $F(P)$ so that the second law ($F = ma$) is satisfied, decompose the acceleration into its angular and translational components, and for each polygon P_i in P define the constraints:

$$weight(i) + \sum_{n \in nodes(P_i)} f_n = mass(i) A_{cm(i)} \tag{4.58}$$
$$\sum_{n \in nodes(P_i)} f_n \times \rho_n = MIcm(i)\theta_i'(t)k, \tag{4.59}$$

where ρ_n is the vector $n - cm(i)$, and k is the unit vector orthogonal to the plane. The first constraint requires that the sum of the forces on P_i equal the mass of P_i times its translational acceleration, while the second requires that the resultant of the forces on P_i taken about its center of mass equal its moment of inertial about the center of mass times P_i's rotational acceleration.

Since P is initially stationary, if P is unstable, then the acceleration $A_{cm(i)}$ must be non-zero for some P_i. To be consistent with a the feasibility constraints for non-intersecting motion, $A_{cm(i)}$ must be consistent with a feasible virtual motion V, i.e., $A_{cm(i)} = V_{cm(i)}(t)$, for all polygons P_i.

Theorem 4.8 *Infinitesimal instability for configurations of polygons with friction is NP-complete.*

Proof By Theorem 4.7, infinitesimal instability for configurations of polygons with friction is NP-hard. Thus it remains to show that it is in NP.

A configuration with friction P is infinitesimally unstable only if there is a pair (F, V) that satisfies the feasibility constraints for configurations with friction described above, such that $W'(0) > 0$.

The system modeling the stability of a configuration of polygons with friction contains the constraints that follow. Each can be formulated as a linear constraint with coefficients having size polynomial in the size of the input. The proof shows that the constraints can be formulated as an instance of the minimum scalar product problem (Problem A.2) that has size polynomial in the size of P. Since Problem A.2 is in NP, this shows that instability for configurations of polygons with friction is in NP.

1. The infinitesimal feasibility constraints for the infinitesimal motion V. Section 3.2.1 showed that the constraints to ensure that V is feasible are linear inequalities with rational coefficients that are small enough.

2. $F = ma$. Equation 4.58 can be formulated as a polynomial number of linear inequalities (two inequalities for each polygon in P), each containing a linear (in the number of nodes of P) number of variables, the coefficients of which are part of the input, and hence small enough.

3. $F_{AB} = -F_{BA}$. Thus if n_1 is a node of P_i and n_2 is a node of P_j, then $f_{n_1} = -f_{n_2}$. This is a linear equality with rational coefficients.

4. If the infinitesimal motion would cause a point of contact to disappear, then the force at that point is zero. this is enforced by the constraint

$$(V_i^j \cdot N(e_1) > 0 \wedge V_i^j \cdot N(e_2) > 0) \Rightarrow f_{n_i} = 0. \tag{4.60}$$

   Equation 4.60 is equivalent to

$$(V_i^j \cdot N(e_1) \leq 0) \vee (V_i^j \cdot N(e_2) \leq 0) \vee (f_{n_i} = 0), \tag{4.61}$$

   which is a formula with linear inequality base terms combined with $\vee$, and thus acceptable input to Problem A.2.

5. If the reaction force at p is such that $V_i^j = 0$, then the force must satisfy Condition 4.2. Thus the following constraint

$$\begin{aligned} V_i^j = 0 \Rightarrow f_{n_i} &= f_{e_1} + f_{e_2} \\ &\wedge \ \mu_{ij} f_{e_1} \cdot e_1 < f_{e_1} \cdot N(e_1) \\ &\wedge \ \mu_{ij} f_{e_2} \cdot e_2 < f_{e_2} \cdot N(e_2), \end{aligned}$$

   which is equivalent to

$$\begin{aligned} V_i^j \neq 0 \vee f_{n_i} &= f_{e_1} + f_{e_2} \\ &\wedge \ \mu_{ij} f_{e_1} \cdot e_1 < f_{e_1} \cdot N(e_1) \\ &\wedge \ \mu_{ij} f_{e_2} \cdot e_2 < f_{e_2} \cdot N(e_2). \end{aligned}$$

   Each of the constraints is a linear constraint with coefficients that are part of the input.

6. If V_i^j is tangent to an edge e_i, then the frictional force f_{n_i} must oppose V_i^j.

$$V_i^j \cdot N(e_i) = 0 \;\Rightarrow\; \mu_{ij} f_{e_1} \cdot e_1 = f_{e_1} \cdot N(e_1)$$
$$V_i^j \cdot f_{e_i} \;<\; 0.$$

Each of these constraints is a linear constraint with coefficients that are part of the input.

Thus the formulation of the constraints satisfies Lemma A.4. The objective function is the formulation of the infinitesimal work $W'(t)$.

The formula for $W'(t)$ is

$$W'(t) = \sum_{n \in P} f_{n_i}(t) \cdot V_i^j + \sum_{1 \leq i \leq n} (weight(i) \cdot V_{cm(i)}(t) + \theta_i MIcm(i)). \tag{4.62}$$

Thus the stability problem can be viewed as the problem of maximizing Equation 4.62 subject to the linear constraints F.

Equation 4.62 can be formulated as a pair of vectors (V_1, V_2) such that (V_1, V_2, F) is an instance of Problem A.2 as follows.

Let

$$V_1 = (f_1, \ldots, f_m, 0, \ldots, 0) \tag{4.63}$$

be a vector of length $2m$, where $f_1, \ldots, f_m$ are the the forces of $F(P)$, and let

$$V_2 = (0, \ldots, 0, v_1, \ldots, v_m) \tag{4.64}$$

be a vector of length $2m$, where $v_1, \ldots, v_m$ are the velocity vectors at the nodes of P. Then $V_1 \cdot V_2$ is a formulation of Equation 4.62. Thus the problem can be formulated as an instance of Problem A.2, such that (V, C) that satisfies Lemma A.5, and hence is in NP. ∎

Corollary 4.6 *Stability for configurations of polygons with friction is CO-NP-complete.*

4.5.3 Potential stability for configurations with friction

The system used to model infinitesimal stability for configurations with friction is similar to that used to model frictionless stability in Section 4.4.3. The system is derived by positing that the system is stable, i.e., that the acceleration $a = 0$, and hence that since $F = ma$, $F = 0$. The configuration is infinitesimally stable if there is a solution to the system of equations that model the configuration.

The only difference between the system used for configurations with friction and that of Section 4.4.1 is that the feasibility constraints for contact forces are relaxed to allow the direction of force lie with in a "cone of friction" consistent with Condition 4.2. Thus, for any node N in contact, the reactive force f_{n_i} must satisfy:

$$f_{n_i} = f_{e_1} + f_{e_2} \tag{4.65}$$
$$\wedge \;\; \mu_{ij} f_{e_1} \cdot e_1 < f_{e_1} \cdot N(e_1) \tag{4.66}$$
$$\wedge \;\; \mu_{ij} f_{e_2} \cdot e_2 < f_{e_2} \cdot N(e_2) \tag{4.67}$$

This is a set of linear inequalities with rational coefficients, as shown in Section 4.5.2.

Theorem 4.9 *Potential stability for configuration of polygons with friction is in P.*

Proof The system that models infinitesimal stability for polygons with friction is a linear program with rational coefficients of size polynomial in the input configuration P. Since linear programming is in P, infinitesimal stability is as well. ∎

Corollary 4.7 *Instability for configurations of polygons with friction is in P.*

Proof Instability is the complement of infinitesimal stability, and P = CO-P. ∎

4.5.4 Complexity of stability problems

This section is a recapitulation of the results of this chapter. The stability problem is CO-NP-hard, for both the frictionless and friction model. Assuming that P $\neq$ NP, this means that there is no deterministic polynomial time algorithm for determining whether a configuration of polygons is guaranteed to be stable, either for frictionless polygons or for polygons with friction. Section 4.6 describes a restricted class of configurations for which there are efficient algorithms for determining whether configurations of polygons in this class are infinitesimally stable or not.

The instability problem is in P for both models. Thus there is an efficient algorithm for determining whether a configuration of polygons is guaranteed to be unstable. This same result shows that it is possible to determine in deterministic polynomial time that a configuration is potentially stable.

4.6 Determined configurations

The results of the previous sections show that the problem of determining the stability of a given configuration of polygons is in CO-NP, even for convex polygons having frictionless surfaces.

Many problems that are provably computationally difficult (e.g., problems that are NP-complete) can, in practice, be successfully "solved." For some intractable problems good approximation or probabilistic algorithms exist, for other problems heuristics can be successful. It is sometimes possible to restrict the domain to allow efficient algorithms for an interesting subclass of the original problem. This section describes such a restricted domain, the *determined* configurations, and presents a polynomial algorithm to test the infinitesimal stability of frictionless configurations in this domain.

4.6.1 Determined configurations

This section describes a restriction on the way that polygons can be in contact that ensures that the system modeling the mobility of a configuration can be solved in deterministic polynomial time. This shows there are polynomial algorithms for both determined mobility and determined frictionless stability.

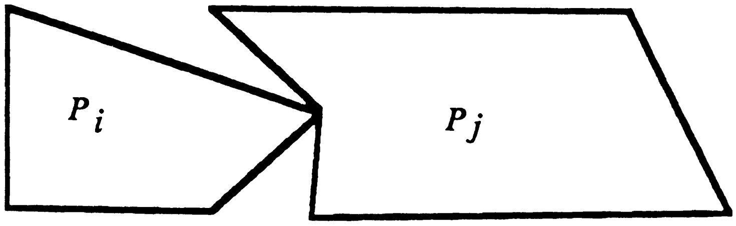

Figure 4.3: Node-node contact where one polygon forms an interior angle greater than or equal to π

The system of constraints described in Section 4.4.1, is a formula satisfying Lemma A.4. That is, it is a formula constructed of base terms that are linear inequalities, which are combined with the connectives **AND** and **OR**. If the class of configurations can be restricted so that, for each configuration P, no **OR** connectives appear in the formula modeling its stability, the formula describes a system that is a linear program. Since LP is in P, the problem of stability for these configurations can be solved in deterministic polynomial time.

An examination of the construction of Constraints(P) shows that the only place that **OR**'s occur is in DirExt(p, P_i, V), (Definition 3.1), and that they occur only when the interior angle formed at a node is less than π.

Theorem 4.10 *Suppose $p = n_i = n_j$ is a point of contact between two polygons P_i and P_j, and that one of the interior angles formed at n_i and n_j is greater than or equal to π. Suppose without loss of generality that the interior angle ϕ_i formed at n_i is such that $\phi_i \geq \pi$. Then DirExt(n_j, P_i, V_{n_j}) $\Rightarrow$ DirExt(n_i, P_j, V_{n_i}). (Recall that V_{n_j} is the relative velocity of n_j with respect to P_i, and similarly that V_{n_i} is the relative velocity of n_i with respect to P_j.)*

Proof Figure 4.3 shows a point of contact that satisfies the statement of the theorem. Obviously the angle $\phi_j < \pi$, and thus by Definition 3.1

$$\mathrm{DirExt}(n_i, P_j, V_{n_i}) = (V_{n_i} \cdot N(e_{j1}) \geq 0) \vee (V_{n_i} \cdot N(e_{j2}) \geq 0). \tag{4.68}$$

Similarly, since $\phi_i \geq \pi$,

$$\mathrm{DirExt}(n_j, P_i, V_{n_j}) = (V_{n_j} \cdot N(e_{i1}) \geq 0) \wedge (V_{n_j} \cdot N(e_{i2}) \geq 0). \tag{4.69}$$

The constraints on the direction of the relative motion of n_i with respect to P_j are shown in Figure 4.4, and Figure 4.5 shows the feasible directions for n_j with respect to

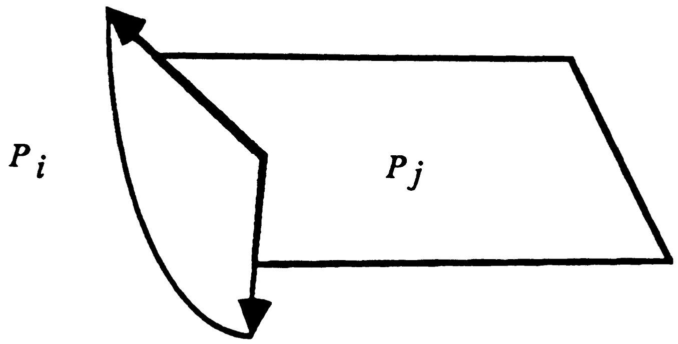

Figure 4.4: Feasible directions for the relative velocity of n_i with respect to P_j

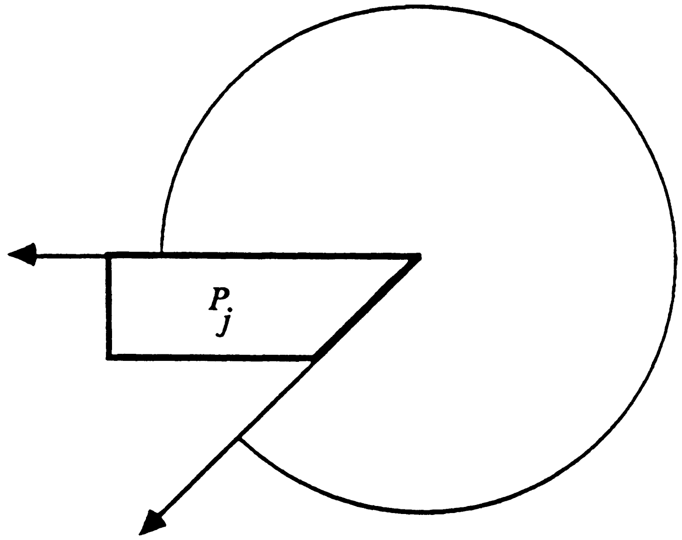

Figure 4.5: Feasible directions for the relative velocity of n_i with respect to P_j

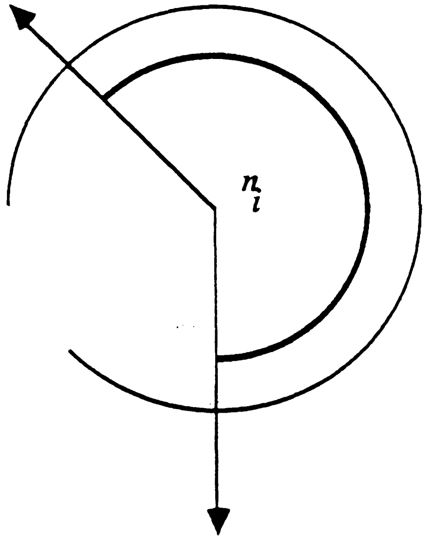

Figure 4.6: Feasible directions for the velocity of n_i

P_i. By Lemma A.3, $V_{n_i} = -V_{n_j}$. Thus the velocity of V_{n_i} must satisfy

$$(-V_{n_i} \cdot N(e_{i1}) \geq 0) \wedge (-V_{n_i} \cdot N(e_{i2}) \geq 0). \tag{4.70}$$

The feasible directions for V_{n_i} are shown in Figure 4.6. Since the set of velocities that satisfy this constraint is contained in the set of velocities that satisfy DirExt(n_i, P_j, V_{n_i}), it follows that DirExt(n_i, P_j, V_{n_i}) is redundant in Constraints(P). ∎

Definition 4.5 An *undetermined* contact point p is an isolated point of contact that is the intersection of two nodes n_i and n_j, (of P_i and P_j, respectively), each of which forms an interior angle of less than π.

A configuration is *determined* if it has no undetermined contact points.

Theorem 4.11 *Infinitesimal mobility for determined configurations is in P.*

Proof Since the OR's are redundant in Constraints(P), it follows that Constraints(P) is a linear program. Thus infinitesimal mobility for determined configurations is in P. ∎

Theorem 4.12 *Infinitesimal stability for determined configurations without friction is in P.*

Proof The system of equations modeling frictionless infinitesimal stability contains no OR's if the configuration is determined. Thus the system modeling the stability of the configuration is a linear program. ∎

Thus, if we are willing to consider only configurations that have no undetermined node-node contact, infinitesimal stability can be determined with linear programming, for which there are both provably efficient algorithms [Ka84], and algorithms that work well in practice, such as the simplex algorithm.

The following theorem shows that the complexity of determining the infinitesimal stability of a given configuration P is $O(2^k Size(P)^r)$, where k is the number of undetermined points of contact, and r is a small constant. Thus, if the configuration has only a few undetermined points of contact, there is an efficient algorithm to determine whether it is infinitesimally stable.

Theorem 4.13 *Suppose a configuration P has k undetermined points of contact. Then the stability of P can be determined in $O(2^k Size(P))$ time, and $O(Size(P))$ space.*

Proof Since only one of the two linear constraints imposed by each undetermined point of contact can be satisfied by any solution to the linear system, it is sufficient to solve the problem for each combination of the **OR** constraints imposed by points of undetermined contact. That is, an algorithm can "guess" which of the two linear constraints will be satisfied, and see if there is a solution. Since there are $O(2^k)$ "guesses," and each verification take $O(Size(P)^n)$, for n a constant, the infinitesimal stability of any configuration of polygons can be tested in $O(2^k Size(P)^n)$ time. Since each of the tests takes space polynomial in $Size(P)$, and little space is required for bookkeeping, stability can be determined in polynomial space.

Chapter 5

Conclusion

This thesis is a preliminary investigation of the computational complexity of testing the stability of configurations of physical objects. It first defines the mobility problem, and shows that mobility for configurations of polygons is NP-hard. These results are used to show that it is CO-NP-hard to determine the stability of configurations of polygons. This chapter describes related and open problems.

5.1 Areas for further research

5.1.1 Generalizations of the model

A three dimensional polyhedral model would allow a stability test for more general physical objects. The complexity of testing the stability of configurations of polygons is a lower bound for testing the stability of polyhedra, since any configuration of polygons can be extruded into a configuration of three dimensional polyhedra. An upper bound on the computational complexity of three dimensional configurations is unknown.

The mechanics of three dimensional polyhedra can be modeled by a set of constraints similar to that of Chapter 4. This suggests the following conjecture.

Conjecture 5.1 *The problem of determining the stability of polyhedra in three dimensions is in CO-NP, and is thus CO-NP-complete.*

A related open problem is formulating a subclass of configurations of polyhedra (corresponding to the *determined* configurations of polygons), for which stability can be tested in polynomial time.

While polyhedra are sufficient to model many physical objects that a robotic planner must reason about, a more general model is required if the behavior of physical objects depends on a non-planar surface. For example, no polyhedral model of a ball is sufficient to test its stability on an incline. Instead, a stability test for configurations of objects defined by parametric equations or constructive solid geometry is required. Stability for more general bodies is largely uninvestigated.

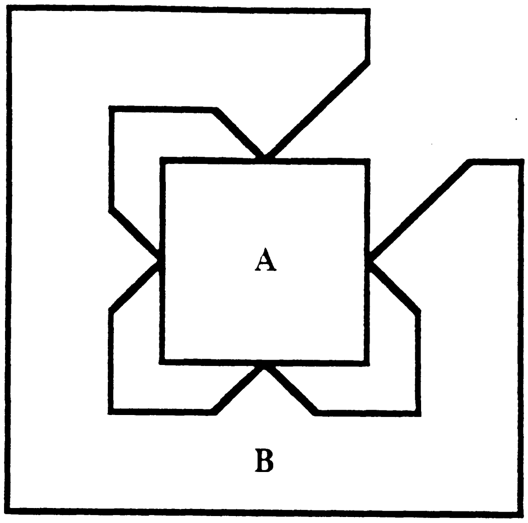

Figure 5.1: A configuration with a feasible velocity but no real feasible motion.

5.1.2 General mobility for polygons in the plane

Chapter 3 showed that infinitesimal mobility is NP-complete. Chapter 4 used this to show that it is CO-NP-hard to determine that a configuration of polygons in the plane is guaranteed to remain stable. The method of virtual work states that a configuration is stable if and only if every feasible assignment of forces has no positive work infinitesimal motion. The characterization of feasible infinitesimal motions presented in this thesis guarantees only that the velocity of any point of contact is not directed towards the interior of another polygon. As Figure 5.1 illustrates, it is possible for a motion to have a feasible velocity, but no real feasible motion. An infinitesimal rotation of A is a feasible velocity, yet the configuration has no real feasible motion.

Conjecture 5.2 *The problem of determining whether a configuration of polygons has a real feasible motion is in NP. In particular, a configuration of polygons satisfying Lemma 2.4 (i.e., the reference frame of each polygon in the configuration is coincident with the plane reference frame) has a real feasible motion if and only if there is a real feasible motion $M(t)$, where each $M_i(t)$ has the form*

$$M_i(t) = (a_i t^2 + b_i t, c_i t^2 + d_i t, e_i t), \tag{5.1}$$

with coefficients a_1, b_i, c_i, d_i, and e_i small enough to be guessed and verified in polynomial time.

5.1.3 Planning stable configurations

Chapter 1 described a more ambitious problem related to stability testing: planning stable configurations. The ability to determine the set of *safe* orientations for a given

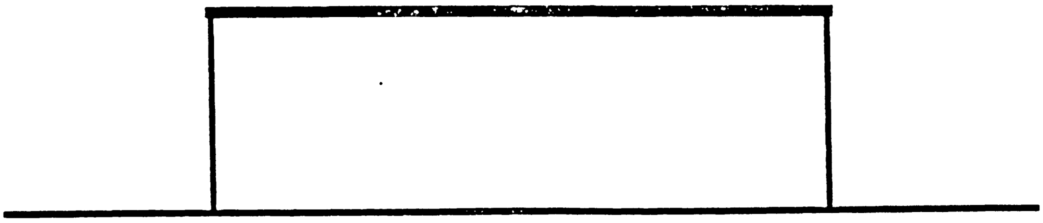

Figure 5.2: A surface capable of supporting force.

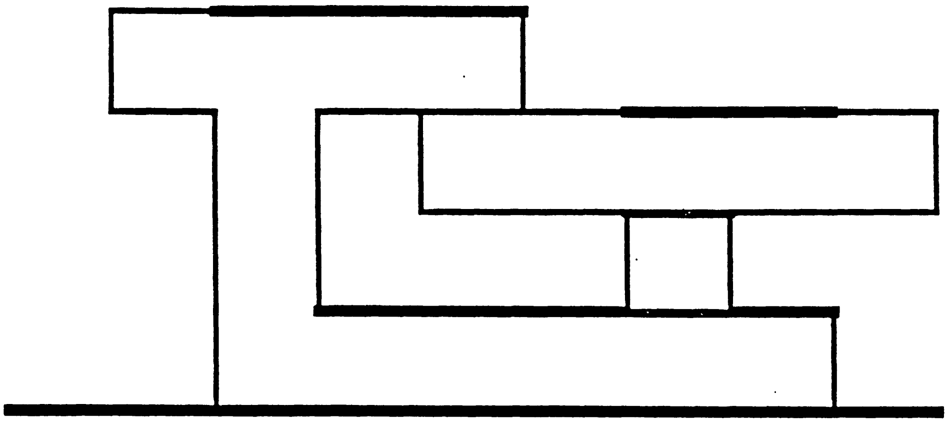

Figure 5.3: Surfaces capable of supporting force.

assembly would allow a system to move an assembly and know its resulting state set. (It would then be possible for the system to determine how to pick up a cup of coffee and move it without spilling.) Additionally, a facility capable of determining the set of surfaces in the workcell on which a given object could be placed would allow a planner to propose configurations that would be guaranteed to be stable.

One promising method for planning stable configurations is to characterize the surfaces of the configuration in terms of their ability to support force. Determining that an object could be placed in a given position would then be reduced to testing whether the resulting reactive forces would satisfy the surfaces of contact.

Conservatively, one could characterize the surfaces in terms of their ability to support an *arbitrary* force As an example, assuming frictionless polygons, the block shown in Figure 5.2 can support an arbitrary force only on its top; force occurring at any other surface will cause it to move. Figure 5.3 is a more complicated example illustrating some of the difficulties in determining such a characterization. The wider edges are those capable of supporting an arbitrary force normal to their surface.

In each step the algorithm would propose a position for a polygon, determine whether it is stable (i.e., satisfies the force constraints for each surface at which it will cause a reactive force to occur), and compute the new regions capable of supporting an arbitrary force. If no polygon is ever removed from the configuration (i.e., if the configuration is monotonic), the set of surfaces capable of supporting force is monotonic and thus easier to compute at each stage. Figure 5.4 shows the force surfaces at each step in the construction of the configuration shown in Figure 5.3.

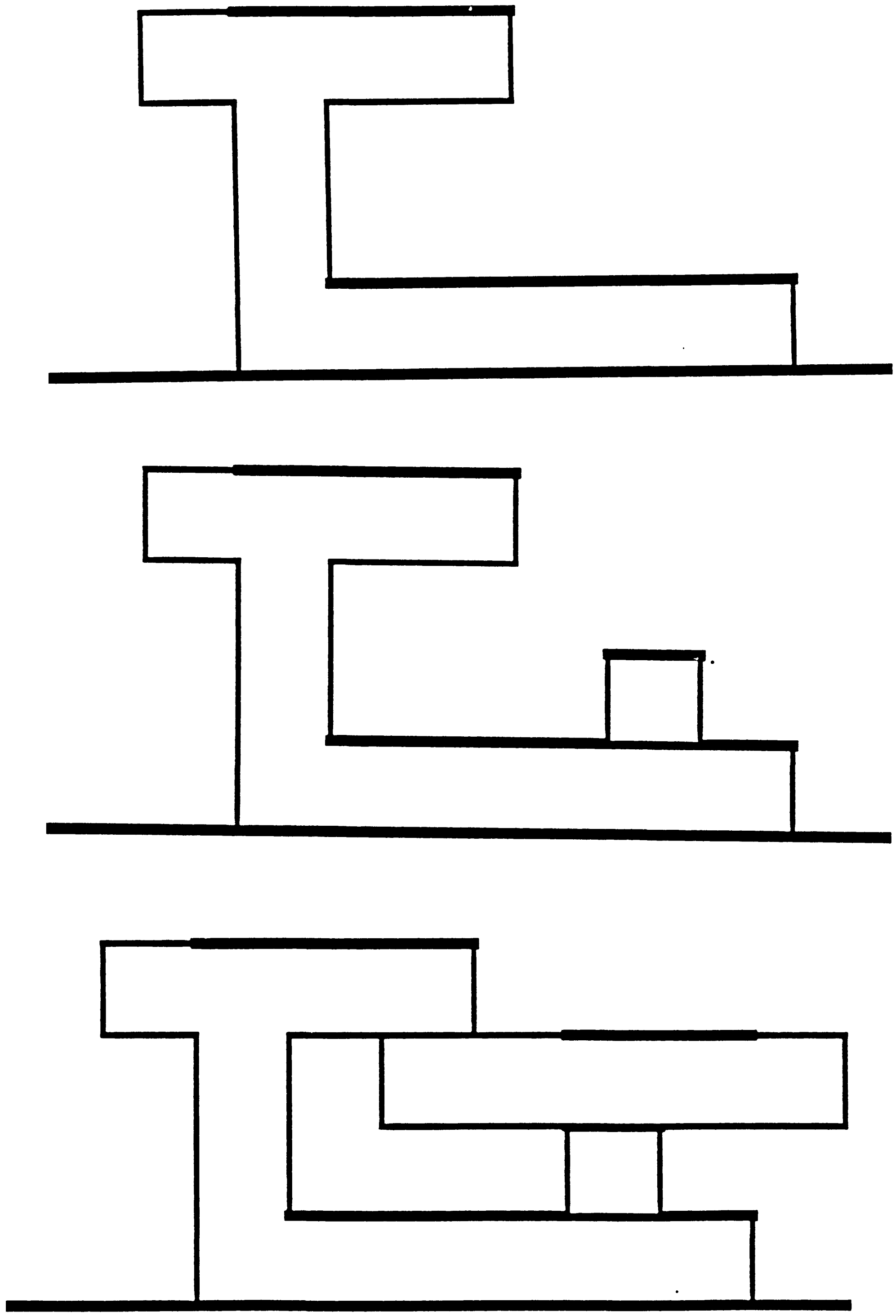

Figure 5.4: Force surfaces in the construction of a configuration.

Appendix A

A.1 Polygons

Definition A.1 A *polygon* is an ordered set of *nodes* $n_1, ..., n_k$. Each node is an ordered pair $n_i = (x_i, y_i)$, which represents the node's displacement from the origin. Assume that the nodes of all polygons are listed in clockwise order.

Definition A.2 The *edges* of a polygon are line segments $e_i = (n_i, n_{i+1})$, for $i = 1, ..., k$. (Where $n_{k+1} = n_1$.)

Definition A.3 A polygon is *simple* if and only if

$$(i < j \wedge e_i \cap e_j \neq \emptyset) \Rightarrow (j = i + 1 \wedge e_i \cap e_j = n_{i+1}).$$

That is, a polygon is defined to be simple if and only if the only intersection among its edges occurs at the endpoints of adjacent edges. Figure A.1 shows examples of simple and non-simple polygons.

All polygons considered in this thesis are simple.

The edges of a (simple) polygon P_i partition the plane into three sets: the points in the *interior* of P_i, those on the *boundary* of P_i, and those on the *exterior* of P_i.

Definition A.4 Define $Boundary(P_i)$ to be the closed set comprising the union of the edges of P_i.

Define $Interior(P_i)$ to be the open set containing those points in the bounded region defined by removing $Boundary(P_i)$ from the plane.

Define $Exterior(P_i)$ to be the open set containing those points not in $Interior(P_i)$ or $Boundary(P_i)$.

This lemma simplifies several proofs in this thesis by limiting the number of cases that need to be considered.

Lemma A.1 *Suppose that P_i is a polygon, and that $p \in Boundary(P_i)$. Then there is a polygon $\widehat{P_i}$ that has a node at p and is equivalent to P_i in the sense that $\widehat{P_i}$ and P_i yield identical Boundary, Interior, and Exterior sets.*

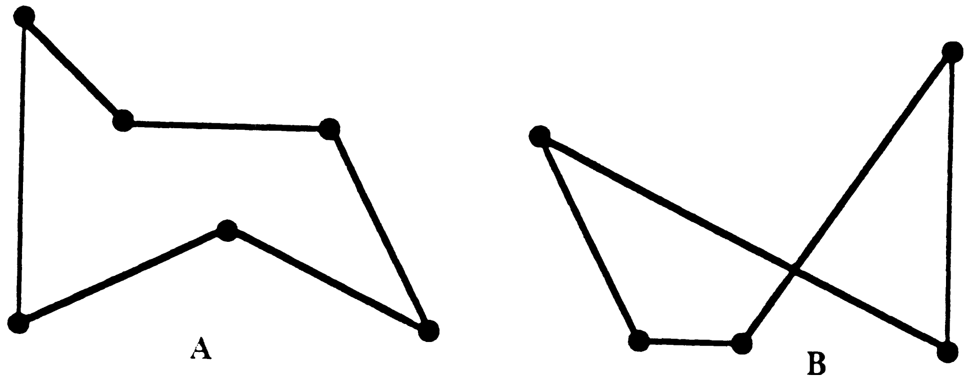

Figure A.1: Simple (A) and non-simple (B) polygons

Proof Suppose a point $p \in Boundary(P_i)$. Then p is either located on a single edge e_j, or at the intersection of two edges e_j and e_{j+1} (i.e., at the node n_{j+1}). Since the statement of the lemma is otherwise trivially satisfied, suppose p is located on an edge e_j of a polygon $P_i = (n_1, \ldots, n_{k_i})$. Define $\widehat{P_i} = (n_1, \ldots, n_j, n_{new}, n_{j+1}, \ldots, n_{k_i})$, where $e_{new1} = (n_j, n_{new})$, and $e_{new2} = (n_{new}, n_{j+1})$. Then $\widehat{P_i}$ is equivalent to P_i in the sense that

$$\begin{aligned} Interior(\widehat{P_i}) &= Interior(P_i), \\ Boundary(\widehat{P_i}) &= Boundary(P_i), \end{aligned}$$

and

$$Exterior(\widehat{P_i}) = Exterior(P_i).$$

∎

Since the nodes of a polygon P_i are listed in clockwise order, the edges of P_i can be considered oriented. Thus it is possible to define a vector that is normal to a given edge e_i, and is directed towards the "exterior side" of e_i as follows.

Definition A.5 Define $N(e_i)$ to be the vector $(y_i - y_{i+1}, x_{i+1} - x_i)$, for $e_i = (n_i, n_{i+1})$. Note that $N(e_i)$ is normal to e_i, is not necessarily of unit length, and is directed towards the "exterior side" of e_i.

A.1.1 Feasible configurations

Definition A.6 Define

$$\begin{aligned} NoIntersect(P_i, P_j, C) &= (p_i \in Interior(P_i) \wedge p_j \in Interior(P_j)) \\ &\Rightarrow \mathrm{Pos}(p_i, l_i) \neq \mathrm{Pos}(p_j, l_j). \end{aligned}$$

Definition A.7

$$\begin{aligned}&IsFeasible(P, C) = \\ &\qquad (\forall i, j : 1 \leq i < j \leq k)(NoIntersect(P_i, P_j, C)).\end{aligned}$$

$IsFeasible(P, C)$ is true if and only if C is positionally feasible for P.

Definition A.8 Define $Feasible(P)$ to be the set of all configurations C that are feasible for P. That is,

$$Feasible(P) = \{C | IsFeasible(C, P)\}.$$

The following lemma shows that a configuration is feasible if and only if the boundary of no polygon intersects the interior of another.

Lemma A.2

$$IsFeasible(P, C) = (\forall i, j) : (Boundary(P_i) \cap Interior(P_j) = \emptyset).$$

Proof The interiors of two polygons can intersect if and only if either

1. some polygon P_i is completely contained in another polygon P_j, in which case

   $$Boundary(P_i, l^i) \cap Interior(P_j, l^j) = Boundary(P_i, l^i),$$

   which is not empty, or

2. there is some $Interior(P_i, l^i)$ that intersects but is not completely contained in $Interior(P_j, l^j)$. Clearly both

   $$Boundary(P_i, l^i) \cap Interior(P_j, l^j) \neq \emptyset,$$

   and

   $$Boundary(P_j, l^j) \cap Interior(P_i, l^i) \neq \emptyset.$$

∎

Definition A.9 For a polygon P_i in configuration C, define $Interior(P_i, l_i)$ to be the interior of P_i in position l_i. That is

$$Interior(P_i, l_i) = \{Pos(p, l_i) | p \in Interior(P_i)\}.$$

$Boundary(P_i, l_i)$ and $Exterior(P_i, l_i)$ are defined analogously.

Definition A.10 For a given configuration of polygons (P, C), distinct polygons P_i and P_j are said to be *positionally feasible* if there is no intersection between their interiors. Define

$$NoIntersect(P_i, P_j, C) = (Interior(P^i, l^i) \cap Interior(P^j, l^j) = \emptyset).$$

A.2 Motion

A.2.1 Translations

Definition A.11 $M_i(t) = (x_i(t), y_i(t), \theta_i(t))$ is a *translation* if $\theta_i(t) = c$, for c a constant.

$M = (M_1, \ldots, M_k)$ is a *translational motion* if, for $1 \leq i \leq k$, each M_i is a translation.

A vector $V = (c_1, c_2, c_3)$ is a *translational velocity* if $c_3 = 0$.

$M = (M_1, \ldots, M_k)$ is a *constant rate translational motion* if, for $1 \leq i \leq k$, each $M_i(t) = M_i(0) + V_i t$, for some translational velocity V_i.

A.2.2 Nonzero initial velocity proof

Notation A.1 *$\frac{d^j}{dt}M$ denotes the j-th derivative of M with respect to t. $M'(t)$ denotes the first derivative of M with respect to t*

The proof of Lemma 2.2:

Proof Let $a = \max\{b | (\forall 0 \leq c \leq b) M(0) = M(c)\}$. Then $0 \leq a < 1$. Then since M is continuous there is some i such that $M_i(a) \neq M_i(a + \epsilon)$, for all $\epsilon > 0$ sufficiently small. Thus there exists $k > 0$ such that $\frac{d^k}{dt}M_i(a) \neq (0, 0, 0)$. Let j be the least such k.

By Lemma 2.1, there is a real feasible motion $\widehat{M}(t) : [0, 1]$. that is path equivalent to $M(t) : [a, 1]$ Then $\widehat{M}(t) : [0, 1]$ is a real feasible motion such that $\frac{d^j}{dt}M_i(0)$ is not $(0, 0, 0)$.

Let $\widetilde{M}(t) = \widehat{M}(t^{1/j})$. To verify that $\widetilde{M}_i'(0) \neq (0, 0, 0)$, consider the Taylor expansion of $\widehat{M}_i'(t)$ about the point $t = 0$,

$$\widehat{M^i}(t) = \sum_{k \geq 0} \frac{1}{k!} \frac{d^k}{dt} \widehat{M}_i(0) t^k \tag{A.1}$$

By hypothesis, the first $j - 1$ derivatives are zero at $t = 0$, so

$$\widehat{M^i}(t) = \sum_{k \geq j} \frac{1}{k!} \frac{d^k}{dt} \widehat{M}_i(0) t^k \tag{A.2}$$

Therefore,

$$\begin{aligned} \widetilde{M^i}(t) &= \widehat{M}_i(t^{1/j}) = \sum_{k \geq j} \frac{1}{k!} \frac{d^k}{dt} \widehat{M}_i(0) (t^{1/j})^k && \text{(A.3)} \\ &= \sum_{k \geq j} \frac{1}{k!} \frac{d^k}{dt} \widehat{M}_i(0) t^{k/j} && \text{(A.4)} \end{aligned}$$

Note that $\widehat{M}(t)$ is defined at $t = 0$. Thus $\widetilde{M}(t) : [0, 1]$ is feasible, since for $0 \leq t \leq 1$ and $k > 0$, $0 \leq t^{k/j} \leq 1$.

Differentiating gives

$$\widetilde{M^i}'(t) = \sum_{k \geq j} \frac{k/j}{k!} \frac{d^k}{dt} \widehat{M}_i(0) t^{(k/j)-1} \tag{A.5}$$

At $t = 0$, this becomes

$$\frac{1}{j!}\frac{d^j}{dt}\widehat{M}_i(0), \tag{A.6}$$

since all other terms in the series have a power of t in them, it follows that

$$\widetilde{M^i}' = \frac{1}{j!}\frac{d^j}{dt}\widehat{M}_i(0) \neq (0,0,0). \tag{A.7}$$

Thus $\widetilde{M}(t) : [0,1]$ is a real feasible motion that is path equivalent to $M(t) : [0,1]$, and has a nonzero velocity at $t = 0$. ∎

A.2.3 Inverse velocity lemma

Lemma A.3 *Suppose M is initially zero, and that p coincides with nodes n_i and n_j of P_i and P_j, respectively. Let V_{n_i} be the relative velocity of n_i with respect to P_j, and V_{n_j} be the relative velocity of n_j with respect to P_i. Then $V_{n_i} = -V_{n_j}$.*

Proof

The proof is a verification of the formulae of V_{n_i} and V_{n_j}. Note that

$$Pos(n_i, M_i(0)) = Pos(n_j, M_j(0)) = p. \tag{A.8}$$

Let $p = (x_p, y_p, 1)$. By Equation 2.11, the absolute velocities of n_i and n_j are

$$[x_p, y_p, 1]\begin{bmatrix} 0 & \theta_i'(0) & 0 \\ -\theta_i'(0) & 0 & 0 \\ x_i'(0) & y_i'(0) & 1 \end{bmatrix}, \tag{A.9}$$

and

$$[x_p, y_p, 1]\begin{bmatrix} 0 & \theta_j'(0) & 0 \\ -\theta_j'(0) & 0 & 0 \\ x_j'(0) & y_j'(0) & 1 \end{bmatrix} \tag{A.10}$$

respectively.

Thus the relative velocity of n_i with respect to that of P_j is

$$\begin{aligned} V_{n_i} &= pM_i'(0)M_j^{-1\prime}(0) \\ &= [x_p, y_p, 1]\begin{bmatrix} 0 & \theta_i'(0) - \theta_j'(0) & 0 \\ \theta_j'(0) - \theta_i'(0) & 0 & 0 \\ x_i'(0) - x_j'(0) & y_i'(0) - y_j'(0) & 1 \end{bmatrix}. \end{aligned}$$

Exchanging i and j in Equation A.11 negates the matrix. Thus $V_{n_i} = -V_{n_j}$.

∎

A.3 Homogeneous transformations

The homogeneous representation of an object in n-space is an object in $(n+1)$-space, in which every point $(x_1, \ldots, x_{n+1})$ in the $(n+1)$-dimensional space maps to the point $(x_1/x_{n+1}, \ldots, x_n/x_{n+1})$ in n-space. Thus if $\mathcal{H}(p)$ is a point in the homogeneous space that represents the point p in n-space, any scalar multiple of $\mathcal{H}(p)$ also represents p. That is, the map from n-space to $n+1$ homogeneous space is one-many. Using homogeneous coordinates allows a transformation (such as that between points expressed with respect to a polygon's coordinate frame and points expressed with respect to the plane origin) to be treated as a single mathematical entity having well understood algebraic properties. In this thesis I use the homogeneous representation for notational convenience, and the ordinary representation for mathematical operations such as differentiation.

For points in the plane, the point $\mathcal{H}(p) = (x, y, w)$ in homogeneous coordinate space represents the point $p = (x/w, y/w)$ in the plane. In this thesis it is always the case that $w = 1$, and thus that the point $p = (x, y)$ is represented by the point $(x, y, 1)$ in homogeneous space.

If $l_i = (x_i, y_i, \theta_i)$ is the position of an oriented polygon P_i, then the transformation from points expressed with respect to P_i's reference frame to the plane reference frame is given by the matrix

$$\mathcal{H}(l_i) = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -\sin\theta_i & \cos\theta_i & 0 \\ x_i & y_i & 1 \end{bmatrix}. \tag{A.11}$$

Definition A.12 Suppose a point $p = (x, y, 1)$ is represented with respect to P_i's reference frame. Let $\mathrm{Pos}(p, l_i)$ denote the position of p in the plane. That is, $\mathrm{Pos}(p, l_i)$ is the position of p with respect to the plane reference frame. $\mathrm{Pos}(p, l_i)$ is given by the equation

$$\mathcal{H}(\mathrm{Pos}(p, l_i)) = [x, y, 1] \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -\sin\theta_i & \cos\theta_i & 0 \\ x_i & y_i & 1 \end{bmatrix}. \tag{A.12}$$

Thus in ordinary coordinates

$$\mathrm{Pos}(p, l_i) = (-\sin(\theta_i)y + \cos(\theta_i)x + x_i, \cos(\theta_i)y + \sin(\theta_i)x + y_i). \tag{A.13}$$

A.4 Complexity techniques

This section defines problems used in Chapter 3 and Chapter 4 to show that various geometric problems are in NP.

Notation A.2 *A function $f(t)$ is* polynomial in *$g(t)$ if $f(t)$ is $O((g(t))^n)$, for some $n > 0$. Define the notation $f(t) \overset{P}{\preceq} g(t)$ to denote "$f(t)$ is polynomial in $g(t)$." Thus $f(t) \overset{P}{\preceq} g(t)$ if and only if $(\exists n : n > 0)$ such that $f(t)$ is $O((g(t))^n)$.*

A.4.1 Boolean formulae with linear inequality base terms

Several of the problems in this thesis can be reduced to the problem of minimizing a linear objective function subject to feasibility constraints defined by a boolean expression having linear inequalities as base terms.

Problem A.1 (Linear boolean expression) *Given* (V, F, O)*, where*

1. V *is a set of* variables.
2. F *is a formula using the connectives* **AND** *and* **OR**, *and base terms* c_i, *where each* c_i *is a linear inequality in* V *with rational coefficients,*
3. O *is a linear* objective function *of* V *with rational coefficients*

the solution to the linear boolean expression problem is:

1. **FALSE** *if there is no solution to* F.
2. X *such that* $F(X) \wedge O(X) = \min\{O(V)\}$ *otherwise.*

That is, the solution is an assignment X *of rational numbers to the variables* V *that satisfies the formula* F *(if there is one) and minimizes* O.

Lemma A.4 *The linear boolean expression problem is in NP.*

Proof Suppose that a given instance of the linear boolean expression problem (V, F, O) has a solution X. Let $C = (c_1, \ldots, c_k)$ be the constraints (i.e., base term linear inequalities) of F that are satisfied by X. That is, C defines a convex polyhedral region that is the intersection of the half-spaces defined by the base terms of F that evaluate to true for X. Let (V, C, O) be the linear program constructed of C and O. (That is, minimize O subject to C.) Let $\widehat{X}$ be the solution to (V, C, O). Since linear programming is in P [Ka84],

$$Size(\widehat{X}) \stackrel{P}{\preceq} Size((V, C, O)) \stackrel{P}{\preceq} Size((V, F, O)). \tag{A.14}$$

Since $\widehat{X}$ satisfies C, it also satisfies F. Therefore, there is an NP algorithm that guesses C, computes $\widehat{X}$, and verifies that it satisfies F. ∎

A.4.2 Constrained maximum scalar product

The solution to the *constrained maximum scalar product* problem is a pair of vectors that have the maximum scalar product subject a formula constructed of linear base constraints. The constrained maximum scalar product problem is in NP, which is used in Chapter 4 to show that the instability problem for configurations of polygons with friction is in NP.

Problem A.2 (Constrained maximum scalar product) *Given* (V_1, V_2, F)*, where*

1. V_1 *and* V_2 *are equal sized vectors of* variables,

2. *F is a formula consisting of the connectives* **AND** *and* **OR**, *and base terms c_i, where each c_i is a linear inequality in the variables of $V = V_1 \cup V_2$ with rational coefficients,*

the solution is a pair of vectors

$$X_1 = (x_{11}, \ldots, x_{1k})$$

and

$$X_2 = (x_{21}, \ldots, x_{2k}),$$

where

$$X_1X_2 = (x_{11}, \ldots, x_{1k}, x_{21}, \ldots, x_{2k})$$

is an assignment of rational numbers to the variables of V that satisfies F and maximizes $X_1 \cdot X_2$.

Lemma A.5 *The constrained maximum scalar product problem is in NP.*

Proof The proof is similar to that of Lemma A.4. Suppose there is an assignment $X = X_1X_2$ that satisfies F, and is such that $X_1 \cdot X_2 > 0$. Then as in Lemma A.4, there is a subset of the constraints of F that are satisfied. These constraints define a convex $2k$–dimensional space $S = S_1 \cup S_2$, where

$$S_1 = \{s = (x_{11}, \ldots, x_{1k}, 0, \ldots, 0) | s \in S\}, \tag{A.15}$$

$$S_2 = \{s = (0, \ldots, 0, x_{21}, \ldots, x_{2k}) | s \in S\}. \tag{A.16}$$

By the definition of scalar product, $X_1 \cdot X_2 = |X_1||X_2| \cos(\theta)$, where θ is the angle between X_1 and X_2. Then there is a solution where X_1 and X_2 are at nodes of S_1 and S_2, respectively. This follows from the following argument.

Suppose that $X_1 \cdot X_2$ is maximized subject to F. Then X_1 is on the boundary of S_1, and X_2 is on the boundary of S_2, since otherwise X_1 could be lengthened to reach the boundary, a contradiction. Without loss of generality, suppose X_1 is not at a node of S_1. Then X_1 is on a polyhedral subset H of a hyperplane of dimension h, where $0 < h < k$, the boundaries of which are formed by the constraints of F.

There are two cases:

1. If H is orthogonal to S_2, any point X_{new} on H yields the same scalar product, i.e., $X_1 \cdot X_2 = X_{new} \cdot X_2$. This follows from the fact that $X_{new} - X_1$ is in H and hence orthogonal to X_2. Thus choosing a node on the boundary of H satisfies the theorem.

2. H is not orthogonal to S_2. In this case, it is possible to find a point X_{new} on H such that the $X_{new} \cdot X_2 < X_1 \cdot X_2$. This follows from the fact that no nonzero vector $X_{new} - X_1$ in H is orthogonal to X_2. Since X_1 is not at a node, there is at least one direction that increases the scalar product with X_2., which is a contradiction.

X_1 and X_2 are at nodes, and can be represented with a number of bits polynomial in the size of the constraints. Since there are a polynomial number of nodes in S_1 and S_2, it is possible to guess the satisfied constraints in F, enumerate each pair (n_1, n_2), for $n_1 \in S_1$, $n_2 \in S_2$ to determine whether $n_1 n_2$ satisfies F, and find the pair with the maximum scalar product. ∎

Bibliography

[AHU74] Aho, A.V., Hopcroft, J.E., and Ullman, J.D.
The Design and Analysis of Computer Algorithms.
Addison-Wesley, 1974.

[BGN70] Blum, M., Griffith, A., and Neumann, B.
A stability test for configurations of blocks.
MIT Artificial Intelligence Memo No. 188, February 1970.

[Fa74] Fahlman, S.E.
A planning system for robot construction tasks.
Artificial Intelligence, Vol. 5, 1974.

[GJ79] Garey, Michael R., and Johnson, David S.
Computers and Intractability.
W. H. Freeman, New York, 1979.

[HR62] David Halliday and Robert Resnick
Physics
John Wiley and Sons, 1962.

[Ka84] Karmarkar, N.
A new polynomial-time algorithm for linear programming.
ACM Symp. on Theory of Computing 16 (1984), 302-311.

[Me84] Mehlhorn, Kurt.
Multi-dimensional Searching and Computational Geometry.
Springer-Verlag, 1984.

[SV80] Theory of Machines and Mechanisms.
Joseph Edward Shigley and John Joseph Vicker, Jr.
McGraw-Hill, 1980.

[Sh80] Shames, Irving H.
Engineering Mechanics, Vol. I, Statics.
Prentice Hall, Englewood Cliffs, New Jersey, 1980.