Algorithmica (1993) 10: 201–225

Algorithmica

Orienting Polygonal Parts Without Sensors[1]

Kenneth Y. Goldberg[2]

Abstract. In manufacturing it is often necessary to orient parts prior to packing or assembly. We say that a planar part is *polygonal* if its convex hull is a polygon. We consider the following problem: given a list of n vertices describing a polygonal part whose initial orientation is unknown, find the shortest sequence of mechanical gripper actions that is guaranteed to orient the part up to symmetry in its convex hull. We show that such a sequence exists for any polygonal part by giving an $O(n^2 \log n)$ algorithm for finding the sequence. Since the gripper actions do not require feedback, this result implies that any polygonal part can be oriented *without sensors.*

Key Words. Robotics, Parts feeding, Planning, Grasping, Compliance, Motion planning with uncertainty, Compliant motion planning.

1. Introduction. Manufacturing processes such as injection molding and stamping often produce a stream of parts that must be reoriented before assembly. A *parts feeder* is a machine that orients parts (Figure 1). Although there are many designs for parts feeders, most use hand-crafted mechanisms that depend on the shape of the part. When part geometry changes, the hardware must be mechanically redesigned with a trial-and-error process that can require several months. Such delays are problematic when production runs are short and part geometry changes frequently.

Hence we desire a parts feeder that can be reprogrammed rather than physically modified. Furthermore, we desire an algorithm for automatically generating the appropriate program (or *plan*) from part geometry. In this paper we present a design for a programmable parts feeder for the class of polygonal parts: a planar part is *polygonal* if its convex hull is a polygon. The mechanism is the ubiquitous parallel-jaw gripper (Figure 2). We focus on the algorithm, which takes a description of part geometry as input and generates a plan for controlling the gripper.

We index each gripper action with an angle. Let a **squeeze action**, α, be the combination of orienting the gripper at angle α with respect to a fixed world frame, closing the jaws as far as possible (e.g., with a binary penumatic valve), and then opening the jaws. As illustrated in Figure 3 for a rectangular part, a sequence of

[1] This report describes research conducted in part while the author was a graduate student supported by NSF Grant DMC-8520475 and NASA-Ames Grant NCC 2-463 at the School of Computer Science at Carnegie Mellon University. The author is currently supported by a grant from the Faculty Research Initiation Fund at the University of Southern California.

[2] Institute for Robotics and Intelligent Systems, Computer Science Department, University of Southern California, Los Angeles, CA 90089-0273, USA. goldberg@iris.usc.edu.

Received April, 1991; revised December, 1991. Communicated by Bruce Randall Donald.

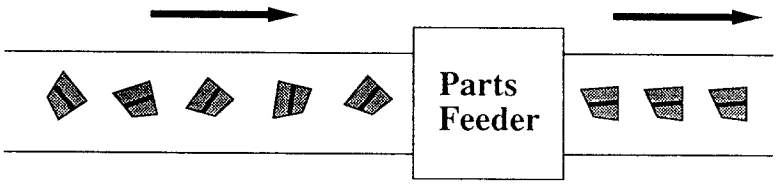

Fig. 1. A parts feeder is a machine that orients parts.

two squeeze actions, $\langle 0, \pi/4 \rangle$, will ensure that the part's major axis is aligned with the gripper. Note that no sensing is required; the sequence orients the part using only mechanical compliance.

In the next section we review related work. We then specify assumptions and analyze the mechanics of squeeze actions before formally defining the problem. In Section 4 we describe a planning algorithm that maps a geometric part description into a sequence of actions for orienting the part. Sections 5 and 6 prove the correctness and completeness of the algorithm, establishing that such a sequence

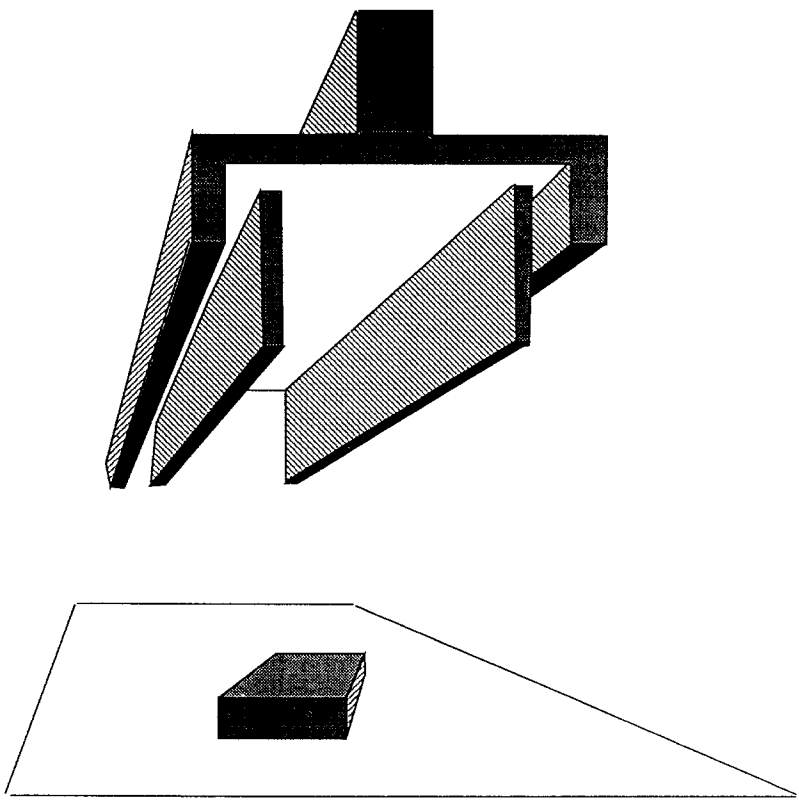

Fig. 2. A parallel-jaw gripper poised above a rectangular part.

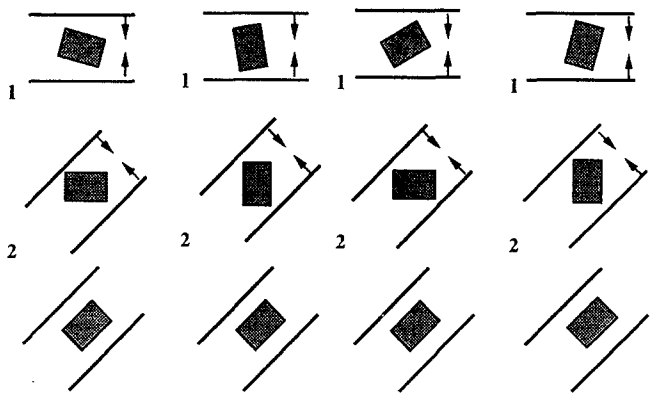

Fig. 3. Four traces, running from top to bottom, of a two-stage plan for orienting a rectangular part. Each squeeze action is indicated with a pair of bold parallel lines. In the absence of friction between the part and the jaws, the part will always rotate into a vertical or horizontal orientation after the first action at 0°. After the second action at 45°, the part is constrained to be in one of two symmetric orientations.

exists for any polygonal part. Section 7 analyzes the time complexity of the algorithm. In Section 8 we show how the algorithm can be extended to the more robust class of *push–grasp* gripper actions.

2. Previous Work.

This method for orienting parts is rooted in two bodies of research: the theory of compliant motion planning and the mechanical analysis of parts feeders.

2.1. *Relation to the Theory of Compliant Motion Planning.*

In *exact* robot motion planning (also known as the *Piano Mover's Problem*) the objective is to find a path (or motion plan) from a known initial configuration (position and orientation of the robot) to a desired final configuration while avoiding a given set of obstacles. For a review of heuristic and algebraic techniques for this problem, see [1]–[4].

In robot motion planning *with uncertainty*, the initial configuration of the robot is not known precisely. For example, suppose we are uncertain about the initial position of a mobile robot in a square room, yet we desire a plan that will move the robot into a corner. In the absence of obstacles, we can achieve this by driving the robot north long enough so that it is stopped by some wall, and then driving it west long enough to guarantee that the robot reaches the northwest corner of the room. Since mechanical compliance is used to reduce uncertainty, this type of solution is known as a *compliant motion plan.*[3]

A geometric approach to compliant motion planning was introduced by Lozano-Perez *et al.* [5]. In this approach the robot is represented as a point in the space of possible configurations. Mechanical properties such as stability,

[3] Also called a *fine motion plan* in [5].

friction, and kinematic constraints are related to geometric conditions in this space. To express uncertainty, both the initial conditions and the desired configuration (the goal) are represented as a *subset* of the configuration space such that the subset is guaranteed to contain the true configuration of the robot. It is also possible to represent uncertainty in the outcome of a nominal commanded motion (e.g., arising from control error) using a one-to-many mapping on the configuration space. A plan, or sequence of motions, creates a composite mapping. A plan that maps the initial set of possible configurations into the goal set is guaranteed to achieve the desired outcome.

One way to generate such a plan automatically is to work backward, known as **preimage backchaining**. Lozano-Perez *et al.* considered a class of planar problems such as inserting a peg into a hole, where peg velocity can only be controlled to within a worst-case error cone. Uncertainty can be reduced by sliding the peg along a surface as in the example above. By transforming the peg into a point in configuration space and projecting a cone that represents velocity error for a nominal commanded motion, it is possible to identify a geometrical region from which the motion is guaranteed to succeed. This region, called the **strong preimage**, then becomes the goal for another motion. If, by chaining backward, we can find a strong preimage that contains the initial set of possible configurations, the sequence of preimages defines a plan that is guaranteed to insert the peg successfully.

Erdmann [6] differentiated between the reachability and recognizability of goal configurations and proposed a restricted class of computable preimages. Donald [7] showed how the approach could be generalized to cope with model error and plan failure and gave an algorithm for planning compliant motions in the plane. Brost [8] developed numerical methods for planar compliant motion planning that are robust to bounded uncertainties in position, orientation, friction, and mass distribution. For other related work, see [9]–[13].

A guaranteed plan does not exist for all problems. For example, the robot's initial configuration may be sufficiently uncertain such that no sequence of actions are guaranteed to achieve the desired goal configuration. Many algorithms for compliant motion planning are *complete* in the sense that they will find a guaranteed plan *if one exists.* Canny and Reif [14] found that deciding if a guaranteed compliant motion plan exists is nondeterministic exponential-time hard. For the problem proposed in this paper, we show that a guaranteed plan always exists.

Both Lozano-Perez [15] and Natarajan [16] noted the similarity between compliant motion planning and the design of parts feeders. The algorithm presented here uses preimage backchaining to find compliant motion plans for a class of planar problems where the system configuration describes the orientation of a polygonal part with respect to the robot gripper. During commanded gripper actions, the part experiences compliant motion, rotating into a stable configuration as the jaws are closed. Subsequently, jaw motion complies to the geometry of the part, halting at this stable configurtion. In a manner analogous to sliding a peg along a sequence of surfaces, the generated sequence of gripper actions constrains the final orientation of the part without recourse to sensors. Mason [17] describes

several other examples where compliant motion plans can be used in lieu of sensors to resolve uncertainty. Such approaches are also known as *open-loop* [18], *sensorless* [19], or *oblivious* [20] manipulation.

2.2. Relation to the Analysis of Mechanical Parts Feeders. Although sensor-based methods such as a binary vision systems can be used to orient parts, sensors introduce additional cost, sampling noise, and require the sensor to be coordinated with a mechanical actuator. Of the methods that do not require sensors, perhaps the most well-known example is the *vibratory bowl feeder*, where parts in a bowl are vibrated with a rotary motion so that they climb a helical track. As they climb, a sequence of pins and cutouts in the track causes parts in all but one orientation to fall back into the bowl for another attempt at running the gauntlet [21], [22]. When part geometry changes, a new helical track is required. Although vibratory bowl feeders are widely used in manufacturing, there are currently no systematic methods for generating track geometry from a description of part shape. Track design is a "black art" performed by specialists through trial and error: an untested part configuration may cause the track to jam during operation. Furthermore, there is no guarantee that an effective track exists for every part.

Hitakawa [23] described a parts feeder that uses an array of nests (silhouette traps) cut into a vibrating plate. The nests and the vibratory motion are designed so that the part will remain in the nest only in a particular orientation. By tilting the plate and letting parts flow across it, the nests eventually fill up with parts in the desired orientation. Although the vibratory motion is under software control, trial and error is required to design the net for each part.

Singer and Seering [24] proposed several designs for parts feeders, one using impact and another where programmed vibration, based on part geometry, was used to drive parts into a stable orientation. Grossman and Blasgen [25] used a vibrating box to ensure that parts fall into one of a finite number of stable orientations under the influence of gravity. They then used a sequence of tactile probes to discriminate among these orientations. All the methods above use vibration, which can be undesirable for fragile parts.

Mason [26] was the first to analyze the role of *pushing* in robot manipulation. Building on results from classical mechanics, he identified a fundamental rule for predicting the direction that a part will rotate as it is pushed in the presence of Coulomb friction. Although the part's exact motion depends on microscopic variations in the support surface, Mason showed that its direction of rotation depends only on the location of the part's center of mass. Other geometric methods for predicting part motion in the presence of frictional contacts were described in [6], [19], [27], [28], and [8]. Mason's rule provided the basis for Brost's [29] *push diagram*, which represents all possible motions of part as it is grasped by a parallel-jaw gripper. Brost showed how to use this diagram to identify single-step grasping actions that are robust to bounded uncertainty in friction and part orientation.

When there is sufficient uncertainty in the initial orientation of a part, more than one action may be required. Several researchers have addressed the problem of finding a *sequence* of actions (a multistage plan) for orienting parts. Given a

finite set of actions, an exhaustive search can be used to consider all possible combinations. One difficulty is that the set of possible actions, often indexed by angle, is uncountably infinite. In the following four studies, the authors used a mechanical analysis based on part geometry to divide an uncountable set of actions into a finite set of equivalence classes. In all cases the objective, as in this paper, is a multistage plan for orienting a polygonal part.

Mani and Wilson [30] considered pushing actions using a single planar fence. Peshkin and Sanderson [31] considered an arrangement of fixed fences where a conveyor belt causes parts to make contact with the fences. Erdmann and Mason [19] considered tilting actions that cause a part to slide into contact with the edges of a rectangular tray. Goldberg and Mason [32] considered grasping actions using a parallel-jaw gripper. In each case the authors developed a mechanical analysis and partitioned the set of possible actions into a finite number of equivalence classes based on part geometry.[4] Each applied a breadth-first exhaustive search to find a sequence of actions guaranteed to produce a unique final orientation of the part. Each method is complete in the sense that it is guaranteed to find such a plan if one exists. However, without a bound on plan length, there is no bound on the computational complexity for any of these search-based planners.

Natarajan [16] was the first to consider the computational complexity of designing parts feeders; he formalized the problem as follows:

> *Given k transfer functions $f_1, f_2, \ldots, f_k$ on a finite set S, find a function f_0 that is a composite of the f_i's such that f_0 is a constant on S, that is,*
>
> $$|f_0(S)| = |\{f_0(s) | s \in S\}| = 1.$$

Although he did not address the problem of finding a physically realistic set of transfer functions based on part geometry, he showed that given k functions, a constant function can be found, if one exists, in time $O(kn^4)$, where n is the size of S. He also identified the restricted class of monotonic transfer functions. Say that a sequence $s_1, s_2, \ldots, s_n$ is **ordered** if each element is encountered exactly once, in that order, when a simple cycle is traced starting with s_1. A function f is **monotonic** if, for any ordered sequence $s_1, s_2, \ldots, s_n$, the sequence $f(s_1), f(s_2), \ldots, f(s_n)$ is ordered. Recently, Eppstein [34] reported an $O(kn^2)$ algorithm for finding such a composite of monotonic functions and showed that finding the shortest composite of nonmonotonic functions is NP-complete.

Erdmann *et al.* [35] reported a computational approach to orienting three-dimensional parts. Given an n-sided polyhedral part resting on a planar table, the objective is to find a sequence of tilting angles for the table that will bring a particular part face into contact with the table (thereby eliminating all but one

[4] Christiansen [33] describes how automated experiments, in lieu of mechanical analysis, can be used for this purpose.

degree of rotational freedom). Assuming that friction is infinite and that the part never rolls off the edge of the table, the authors showed how to construct such a plan, if one exists, in time $O(n^4)$.

In this paper we consider a class of parallel-jaw gripper actions. As described in [32], this uncountable set of gripper actions can be partitioned into $O(n^2)$ equivalence classes based on the geometry of the given n-sided part. However, since every action corresponds to a monotonic transfer function, we could use Eppstein's algorithm to decide if a guaranteed plan exists in time $O(n^4)$. However, since part geometry adds additional structure to our set of transfer functions, we are able to show that a guaranteed plan always exists using a geometric algorithm that finds the shortest guaranteed plan in time $O(n^2 \log n)$. Preliminary descriptions of this work appeared in [18] and [36].

3. Assumptions and Mechanical Analysis.

We assume that:

1. All motion occurs in the plane and is slow enough that inertial forces are negligible. The scope of this *quasi-static* model is discussed in [37] and [38].
2. The gripper consists of two linear jaws arranged parallel to each other.
3. The direction of gripper motion is orthogonal to the jaws.
4. The convex hull of the part can be treated as a rigid planar polygon.
5. The part is presented to the feeder in isolation; we do not address the related problem of isolating parts from a bin (commonly known as *singulating*).
6. The part's initial position is unconstrained as long as it lies somewhere between the jaws. The part remains between the jaws throughout grasping.
7. Both jaws make contact simultaneously (pure squeezing).
8. Once contact is made between a jaw and the part, the two surfaces remain in contact throughout the grasping motion. The action continues until further motion would deform the part.
9. There is zero friction between the part and the jaws.

These assumptions are similar to those made by Brost [29] and by Taylor *et al.* [39], [40], with the exception of assumptions 3, 7, and 9. By restricting gripper motion to be orthogonal to the jaws (assumption 3), we obtain a one-dimensional action space rather than the two-dimensional space considered previously. Assuming simultaneous contact (assumption 7) simplifies the mechanical analysis but is almost never perfectly satisfied; that is, one jaw inevitably makes contact first and pushes for some distance. In Section 8 we show how this assumption can be relaxed by considering the class of *push–grasp* actions.

Friction between the part and the jaws can produce *wedged* configurations where the part is cocked between two vertices [29]. By assuming zero friction (assumption 9), we avoid such configurations. A frictionless gripper can be closely approximated by mounting a linear bearing on one jaw as described in [41].

3.1. *Mechanical Analysis.*

When a polygonal part is grasped with the frictionless gripper, it assumes one of a finite number of "stable" configurations where at least

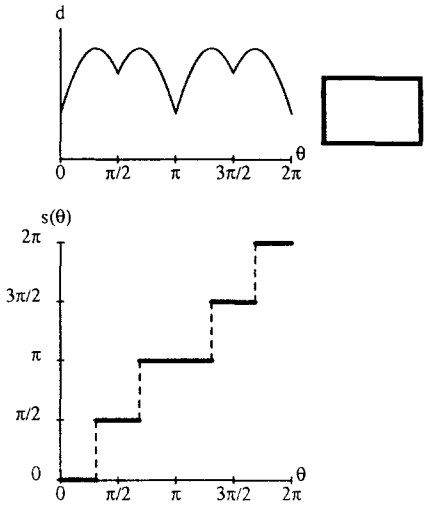

Fig. 4. The diameter function (top) and squeeze function (bottom) for the rectangular part.

one edge of the part's convex hull is in contact with a jaw. Such configurations correspond to local minima in the following function. For a fixed orientation of the part, define the part's *diameter* at direction θ to be the maximum distance between two parallel supporting lines at angle θ. Let S^1 be the set of planar orientations.[5] The **diameter function**, d: $S^1 \rightarrow \Re$ describes how the diameter varies as the parallel lines are rotated around the part (see Figure 16). Note that the diameter is simply the distance separating the jaws of the gripper when both jaws are just touching the polygon. See the Appendix for an $O(n)$-time algorithm for computing the diameter function.

The outcome of a squeeze action can be predicted using the diameter function. We define a transfer function, the **squeeze function**, s: $S^1 \rightarrow S^1$, such that if θ is the initial orientation of the part with respect to the gripper, $s(\theta)$ is the orientation of the part with respect to the gripper after the squeeze action is completed. For a polygonal part, the piecewise-constant squeeze function is derived as follows. All orientations that lie between a pair of adjacent local maxima in the diameter function will map to the orientation corresponding to the enclosed local minimum, i.e., the squeeze function is constant over this interval of orientations. The result is a step function as shown in the bottom of Figure 4.

We assume that all steps in the squeeze function are closed on the left and open on the right. Strictly speaking, the squeeze function has value $s(\theta) = \theta$ at its discontinuities, corresponding to an unstable equilibrium where the part is wedged between two exactly aligned vertices. We could define a squeeze action at angle α to include closing and opening the gripper at angle α followed by closing the gripper at angle $\alpha + \varepsilon$, rotating the gripper by $-\varepsilon$, and then opening the gripper.

[5] Functions representing angles in S^1 are evaluated modulo 2π.

In [42] we show how to find an appropriate ε for any polygonal part such that the combined action has a piecewise constant transfer function where each step is closed on the left and open on the right. In practice, however, mechanical vibration in the gripping mechanism is sufficient to dislodge such wedged configurations, and after the first squeeze action causes the part to rotate into a stable configuration, the plan's margin for error (specified below) allows us to avoid actions that could produce a wedged configuration. For more on this issue, see [20].

Recall that the domain and range for the squeeze function are specified with respect to the gripper. Rotating the gripper corresponds to a shift in the relative orientation of the part. If the part is at angle θ and the action is applied at angle α, the resulting orientation of the part with respect to the gripper will be $s(\theta - \alpha)$.

In what follows we use the term **interval** to refer to a connected subset of S^1. For an interval Θ, let $|\Theta|$ denote its Lebesgue measure. We define an ***s*-interval** to be a semiclosed interval of the form $[a, b)$ such that a, b are points of discontinuity in the domain of squeeze function s. For an s-interval Θ_X, let θ_X refer to its included bounding point. Since there are $O(n)$ discontinuities in the squeeze function, there are $O(n^2)$ unique s-intervals, each of which has nonzero measure. We define the ***s*-image** of a set, $s(\Theta)$, to be the smallest interval containing the following set: $\{s(\theta) | \theta \in \Theta\}$. Note that the s-image of any set will be a closed interval.

3.2. Orienting a Part up to Symmetry. Note that the diameter function has period π due to symmetry in the gripper; rotating the gripper by 180° produces a symmetric arrangement that preserves the diameter. Rotational symmetry in the part also introduces periodicity into the diameter function. This periodicity introduces structure into the squeeze function. We say that a squeeze function **has period** T if, for all θ,

$$s(\theta + T) = s(\theta) + T. \tag{1}$$

For polygons with r-fold rotational symmetry, the squeeze function will have period $T_r = 2\pi/r(1 + r \bmod 2)$. For example, a part with no rotational symmetry ($r = 1$) has a squeeze function with period π. An equilateral triangle has threefold rotational symmetry; its squeeze function has period $\pi/3$. A square has fourfold rotational symmetry; its squeeze function has period $\pi/2$.

Periodicity in the squeeze function gives rise to *aliasing*: any sequence of actions that maps θ to θ' will map $\theta + T$ to $\theta' + T$. This implies that there is no sequence of squeeze actions that can map orientations θ and $\theta + T$ into a single final orientation. For a given part, let T be the smallest period in its squeeze function. We say that a plan orients the convex hull of a part **up to symmetry** if the set of possible final orientations includes exactly $2\pi/T$ orientations that are equally spaced on S^1. For example, for a part with no rotational symmetry, a squeeze plan orients the part up to symmetry if the plan yields exactly two final orientations that are π radians apart. A part with threefold rotational symmetry can be oriented

up to symmetry with a squeeze plan that yields six possible final orientations each $\pi/3$ radians apart.

We define the parts feeding problem as follows:

Given a list of n vertices describing the convex hull of a polygonal part, find the shortest sequence of squeeze actions guaranteed to orient the part up to symmetry.

4. The Algorithm. The algorithm begins with an s-interval whose image is a point. It continues, finding larger and larger s-intervals. When the algorithm terminates, the resulting sequence of s-intervals can be transformed into a sequence of squeeze actions that, in effect, "funnel" the largest s-interval into a unique final orientation. The algorithm follows:

1. Compute the squeeze function.
2. Find the widest single step in the squeeze function and set Θ_1 equal to the corresponding s-interval. Let $i = 1$.
3. While there exists an s-interval Θ such that $|s(\Theta)| < |\Theta_i|$:
   - Set Θ_{i+1} equal to the widest such s-interval.
   - Increment i.
4. Return the list $(\Theta_1, \Theta_2, \ldots, \Theta_i)$.

We illustrate the algorithm using the squeeze function for the rectangular part as reproduced in Figure 5. Since this part has aspect ratio 1.5, let $a = a \tan 2(3, 2)$.

In step 2 of the algorithm the widest single step is found and Θ_1 is set to be the corresponding s-interval on the horizontal axis: $[\pi - a, \pi + a)$. Note that $s(\Theta_1)$ is the unique orientation at angle π.

In step 3 of the algorithm we seek the widest s-interval whose s-image has

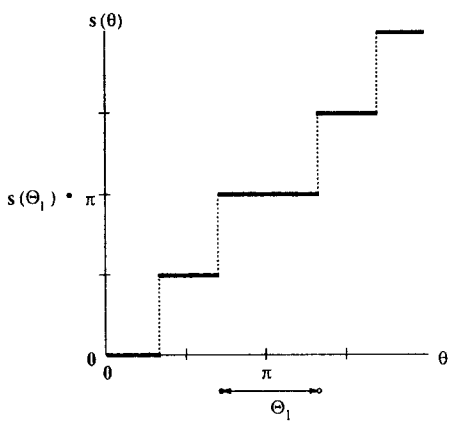

Fig. 5. In step 2 the widest single step in the squeeze function is identified. All the orientations in Θ_1 map into the single final orientation, $s(\Theta_1)$.

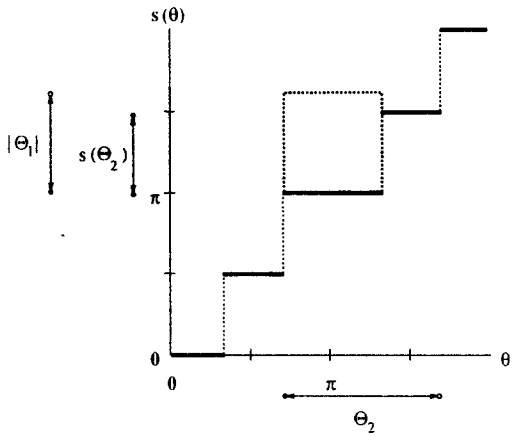

Fig. 6. Illustrating step 3 of the algorithm.

smaller measure than Θ_1. As illustrated in Figure 6, this can be visualized by left-aligning a box of dimensional $|\Theta_1|$ with each step in the squeeze function. If the squeeze function emerges from the right edge of the box, then the s-image of the corresponding s-interval has smaller measure than Θ_1. The largest such s-interval in this case is $\Theta_2 = [\pi - a, 2\pi - a)$. Note that $s(\Theta_2) = [\pi, 3\pi/2]$, $|s(\Theta)_2| = \pi/2 < |\Theta_i| = 2a$.

Continuing in this manner, wider and wider s-intervals are found until the loop terminates. This will occur when $|\Theta_i| = T$, a period of symmetry in the squeeze function as defined in Section 3.2. For the rectangular part, the algorithm terminates with $i = 2$ since $|\Theta_2| = \pi$.

4.1. Recovering the Plan. In this section we show how to recover a plan from a list of s-intervals. Given the list $(\Theta_1, \Theta_2, \ldots, \Theta_i)$, we can generate i plans. For j between 1 and i, a *plan*, ρ_j is a sequence of j squeeze actions, $(a_j, a_{j-1}, \ldots, a_1)$, that collapses all orientations in Θ_j to the unique orientation $s(\Theta_1)$. Thus we can generate a plan to collapse any s-interval in the list. Since $|\Theta_i|$ is the largest s-interval (corresponding to a period of symmetry in the squeeze function), ρ_i will orient the part up to symmetry.

Again, we illustrate using the rectangular part as shown in Figure 6. At the start of the plan, the rectangular part will be in some orientation in Θ_2 (i.e., there is 180° uncertainty in the part's initial orientation). After performing a squeeze action at angle 0, the part's orientation is guaranteed to lie in the s-image $s(\Theta_2)$.

Since $|s(\Theta_2)| < |\Theta_1|$, we could collapse the interval $s(\Theta_2)$ to a point with a single squeeze action at angle 0 *if* $s(\Theta_2)$ could be aligned with Θ_1. Fortunately, this can be accomplished by rotating the gripper by angle $s(\theta_2) - \theta_1 = a$ in the fixed coordinate frame, which has the effect of rotating the part with respect to the gripper by the negative of this angle. Since the orientation of the part in the new gripper frame is guaranteed to lie inside Θ_1, a second squeeze action at angle 0 in the new gripper frame will ensure that the part's final orientation is the unique angle $s(\Theta_1) = \pi$ in the new gripper frame.

Two additional points:

- Since each rotation is expressed relative to the previous gripper frame, each squeeze action must be transformed into the fixed coordinate frame.
- Note that $|s(\Theta_2)| < |\Theta_1|$. Thus we can allow an additional margin of error in part orientation. For the rectangular part, let $\varepsilon_1 = \frac{1}{2}(|\Theta_1| - |s(\Theta_2)|) = a - \pi/4$. By adding an additional increment of ε_1 to the commanded gripper rotation before the second squeeze action, we ensure that the rectangle's orientation with respect to the gripper will lie inside Θ_1 even if the part is perturbed by $\pm\varepsilon_1$.

In general, the i-step plan, $\rho_i = (\alpha_i, \alpha_{i-1}, \ldots, \alpha_1)$, is defined as follows. Set $\alpha_i = 0$. For $j = i - 1$ downto 1,

$$\alpha_j = s(\theta_{j+1}) - \theta_j - \varepsilon_j + \alpha_{j+1}, \tag{2}$$

where $\varepsilon_j = \frac{1}{2}(|\Theta_j| - |s(\Theta_{j+1})|)$.

If the part's initial orientation is contained in the s-interval Θ_i, its final orientation will be the unique angle, $s(\Theta_1) + \alpha_1$. Consider the case where the part's initial orientation is completely unknown. Since Θ_i is a period of symmetry in the squeeze function, it partitions S^1 into $2\pi/T$ disjoint s-intervals of equal size.

Thus for the rectangular part, the plan comprised of a squeeze action at angle 0 followed by a squeeze action at angle $\pi/4$ is guaranteed to orient the part to angle $\pi/4$ if it started out in the range $[-a, \pi - a)$, and to angle $3\pi/4$ if it started out in the range $[\pi - a, -a)$. Note that in Figure 3, orientations $\pi/4$ and $3\pi/4$ are indistinguishable for the rectangle which has rotational symmetry: $r = 2$.

Another example is illustrated in Figures 7 and 8, for the 4-gon described by vertices $(-42, -41)$, $(48, -41)$, $(39, 25)$, $(-34, 59)$.

5. Correctness. To show that the algorithm correctly solves the parts feeding problem, we first show that the resulting plan will orient the part up to symmetry and then show that there is no shorter plan that does so.

The algorithm finds a plan that collapses an s-interval of length T, where T is the smallest period in the part's squeeze function. This implies that the plan orients the part up to symmetry. To see this, note that since $s(\theta + T) = s(\theta) + T$, it follows that, for any plan ρ, $\rho(\theta + T) = \rho(\theta) + T$. Consider a two-stage plan, $\rho_2 = (\alpha_1, \alpha_2)$:

$$\begin{aligned}
\rho_2(\theta + T) &= s(s(\theta + T - \alpha_1) - \alpha_2) \\
&= s(s(\theta - \alpha_1) + T - \alpha_2) \\
&= s(s(\theta - \alpha_1) - \alpha_2) + T \\
&= \rho_2(\theta) + T.
\end{aligned}$$

If a plan collapses all orientations in an s-interval to orientation θ, then it will also collapse all orientations in an s-interval of the same size offset by kT to orientation $\theta + kT$. If the part's initial orientation is unknown, its final orientation

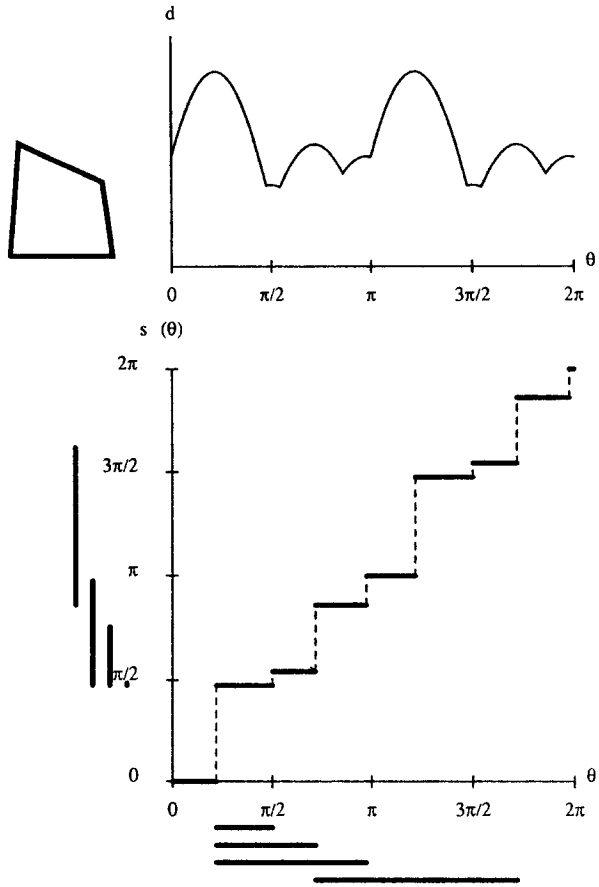

Fig. 7. Geometric analysis of the 4-gon, showing diameter and squeeze functions. The bars under the horizontal axis indicate the sequence of s-intervals found by the algorithm, the bars to the left of the vertical axis indicate the corresponding s-images.

will be one of $2\pi/T$ final orientations equally spaced on S^1. Thus the plan will orient the part up to symmetry as defined earlier. We now prove that the algorithm finds the shortest plan[6] for orienting the part up to symmetry. First we show that it is sufficient to consider only connected subsets of S^1.

LEMMA 1. *Any plan that collapses a set* $\Theta \subseteq S^1$ *will also collapse the smallest (connected) interval containing* Θ.

[6] There may be more than one shortest plan due to ties. We refer to "the" shortest plan although it may not be unique.

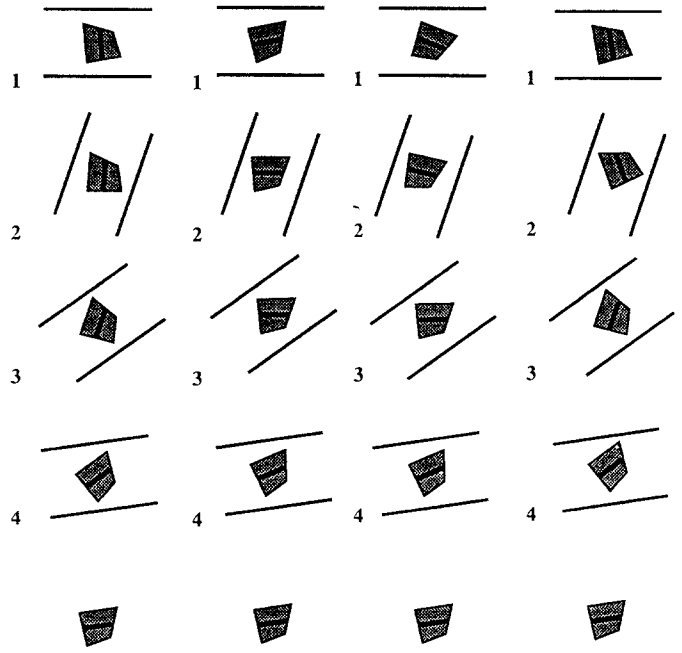

Fig. 8. Four traces of the resulting squeeze plan for the 4-gon.

PROOF. Let Θ' be the smallest (connected) interval containing Θ. Due to monotonicity of the squeeze function, $s(\Theta') = s(\Theta)$. Thus the first squeeze action will produce the same s-image in either case, and we can use the remainder of the plan to collapse Θ' to a point. □

THEOREM 1. *No plan containing fewer steps can orient the part up to symmetry.*

PROOF. Let ρ_i be the i-stage plan found by the algorithm. Suppoe there is a shorter plan, ρ'_j, $j < i$, that orients the part up to symmetry. Let $(\Theta_1, \Theta_2, \ldots, \Theta_i)$ be the list of s-intervals found by the algorithm and let $(\Theta'_1, \Theta'_2, \ldots, \Theta'_j)$ be the corresponding sequence of bounding intervals for plan ρ'_j as constructed in Lemma 1. By definition of the algorithm, $|\Theta_1| \geq |\Theta'_1|$. Since ρ'_j terminates in j steps but ρ_i does not, it must be the case that $|\Theta_j| < |\Theta'_j|$. This means that there exists a point in the sequences of intervals such that $|\Theta_k| \geq |\Theta'_k|$ and $|\Theta_{k+1}| < |\Theta'_{k+1}|$, but this cannot occur by definition of the algorithm. Thus there is no shorter plan that can orient the part up to symmetry. □

6. Completeness

THEOREM 2. *For any polygonal part, we can always find a plan to orient the part up to symmetry.*

PROOF. As described earlier, any polygonal part will generate a piecewise-constant squeeze function, s: $S^1 \to S^1$, where all s-intervals have nonzero measure and $s(\theta + T) = s(\theta) + T$, where T is a period of symmetry. To simplify the problem of wraparound at 0, in this section we extend s to a function on the real line, $s: \Re \to \Re$, that has exactly the same value as before on $[0, 2\pi)$. Elsewhere, it is specified as $s(\theta + T) = s(\theta) + T$.

We prove the claim by showing that, for any such squeeze function, we can always find a sequence of s-intervals, $(\Theta_1, \Theta_2, \ldots, \Theta_i)$, of increasing measure with the condition that Θ_j has larger measure than the s-image of Θ_{j+1}. The trick is to show that, for any piecewise-constant monotonic squeeze function and any s-interval, we can always find a larger s-interval unless it corresponds to a period of symmetry in the squeeze function.

Let h be the measure of some s-interval. Either we can find a larger s-interval whose s-image is smaller than h,

$$\exists \theta, \quad s(\theta + h) - s(\theta) < h, \tag{3}$$

or h is a period of symmetry in the squeeze function:

$$\forall \theta, \quad s(\theta + h) = s(\theta) + h, \tag{4}$$

where the quantifiers range over the interval $[0, T)$.

To understand (3), consider that we have reached a point in the algorithm where the current s-interval is $\Theta_j = [\theta_j, \theta_j + h)$. Formula (3) says that there is some closed interval, $\Theta = [\theta, \theta + h]$, whose s-image is smaller than Θ_j. We can expand Θ (without increasing its s image) by extending it to the right until we reach a discontinuity in the squeeze function. This yields an s-interval whose s-image is smaller than Θ_j. The difference between this s-interval and Θ is an open interval and hence has nonzero measure. Thus this s-interval will have larger measure than Θ_j.

We can also interpret (3) with reference to Figure 6. The formula states that we can always find a position for the lower left-hand corner of the box such that the squeeze function enters on the left edge of the box and exits on the right edge.

To show that, for any such squeeze function and any h, either (3) or (4) must hold, consider the integral of the function $s(\theta + h) - s(\theta) - h$ over the interval $[0, T)$:

$$\int_0^T [s(\theta + h) - s(\theta) - h]\, d\theta = \int_h^{T+h} s(\theta)\, d\theta - \int_0^T s(\theta)\, d\theta - hT \tag{5}$$

$$= -\int_0^h s(\theta)\, d\theta + \int_T^{T+h} s(\theta)\, d\theta - hT \tag{6}$$

$$= -\int_0^h s(\theta)\, d\theta + \int_0^h [s(\theta) + T]\, d\theta - hT \tag{7}$$

$$= -\int_0^h s(\theta)\, d\theta + \int_0^h s(\theta)\, d\theta + hT - hT \tag{8}$$

$$= 0. \tag{9}$$

Since this integral is zero, either there is some point where the function is less than zero ((3) is true) or the function is uniformly zero ((4) is true, i.e., $h = T$).

Hence we can always continue to find larger s-intervals until we reach a period of symmetry in the squeeze function. We have shown earlier that we can transform this sequence of s-intervals into a plan to orient the part up to symmetry. □

7. Complexity. We assume that the following operations take time $O(1)$: finding the distance between two points in the plane, finding the angle between two lines in the plane, comparing two rational numbers, adding or subtracting two rational numbers, and random access of any particular memory location.

THEOREM 3. *For a polygon of n sides, the algorithm runs in time* $O(n^2 \log n)$ *and finds plans of length* $O(n^2)$.

PROOF. Step 1, computing the squeeze function, can be performed in time $O(n)$ as is shown in the Appendix. Since there are $O(n)$ steps in the squeeze function, step 2 takes time $O(n)$. For the loop, recall that a squeeze function defines $O(n^2)$ s-intervals. Suppose we sort these by $|s(\Theta)|$. As we proceed through the loop, we only need to traverse this list once, so that i is $O(n^2)$. Recovering the plan requires $O(i)$ time. Sorting dominates the running time, so the algorithm runs in time $O(n^2 \log n)$. □

8. Push–Grasp Actions. In this section we relax the assumption that both jaws make contact simultaneously by considering the class of *push–grasp* actions first identified by Brost [29]. Let a **push–grasp action**, α, be the combination of orienting the gripper at angle α with respect to a fixed world frame, translating the gripper in direction $\alpha + \pi/2$ for a fixed distance, closing the jaws as far as possible, translating the gripper in direction $-\alpha - \pi/2$ for the same distance, and then opening the jaws.

During the pushing phase, where the gripper translates, the part rotates so that one of its edges is aligned with the pushing jaw before the second jaw makes contact. For this class of actions, we substitute two assumptions for assumption 7 in Section 3:

- The part's center-of-mass is given and the coefficient of friction with the support surface is independent of position and velocity [26].
- The pushing distance is sufficient to align a stable part edge with the pushing jaw before the second jaw makes contact.

A lower bound on the required pushing distance can be computed by considering the smallest disk that covers the part [38]. Although this bound can approach infinity if the part is pushed along a vector from the contact vertex through the part's center-of-mass, we can avoid such actions after the first push–grasp action causes the part to rotate into one of its stable orientations.

The mechanics of pushing can be captured with an analog to the diameter

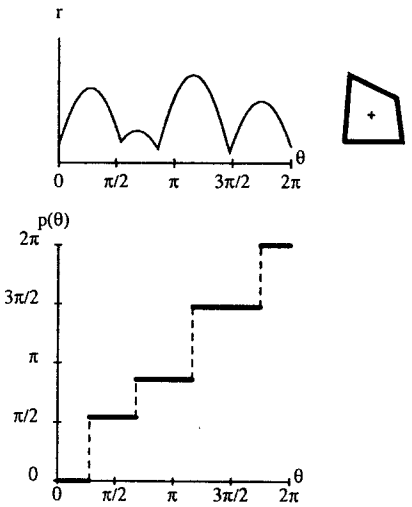

Fig. 9. The radius and push functions for the 4-gon.

function as constructed by Mason [26]. Consider a single line of support, aligned with the x-axis, below the part. Let the part's *radius* at direction 0° be the perpendicular distance from the support line to the part's center of mass. The **radius function**, $r: S^1 \to \Re$, describes how the radius varies as the support line is rotated around the part. See top of Figure 9. By checking each vertex, the radius function for an n-sided polygonal part can be computed in time $O(n)$. When pushed, the part will rotate so as to reduce its radius.

The **push function**, $p: S^1 \to S^1$, maps an initial orientation of the part to a final orientation after the pushing phase. See bottom of Figure 9. It is a step function derived from the radius function in the same way that the squeeze function is derived from the diameter function, i.e., discontinuities occur at local maxima in the radius function. Since the radius function has period 2π unless the part has rotational symmetry, pushing allows us to eliminate the 180° ambiguity in the part's final orientation that is inherent with squeeze actions.

To analyze the mechanics of a push–grasp action, we compose the push function with the squeeze function to get a transfer function, $g: S^1 \to S^1$, that we call the **push–grasp function** as illustrated in Figure 10. We assume that steps are closed on the left and open on the right. As with the squeeze function, this assumption can be formally justified with a minor modification to the definition of the push–grasp action, or we can make a similar argument that metastable orientations can be avoided after the first action. Note that the push–grasp function is also piecewise constant and monotonic, and can be computed in time $O(n)$.

Thus we can use the planning algorithm from Section 4, modifying only step 1 to use the push–grasp function instead of the squeeze function. The correctness, complexity, and completeness of the algorithm can be proved as before. The s-intervals and resulting plan for the 4-gon are shown in Figures 11 and 12.

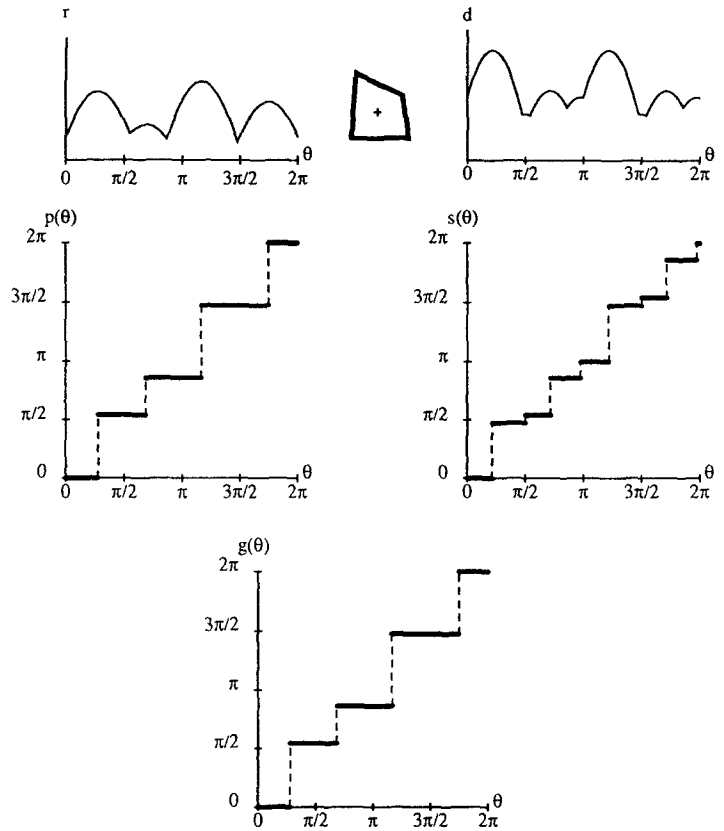

Fig. 10. Push–grasp analysis for the four-sided part. The radius function is in the upper left. Directly below it is the push function. The diameter function is in the upper right. Directly below it is the squeeze function. The push–grasp function is shown at the bottom.

Another example, for a house-shaped part described by the vertices $(-25, -33)$, $(25, -33)$, $(25, 20)$, $(0, 45)$, $(-25, 20)$, is shown in Figures 13–15.

9. Discussion. We have described an algorithm for generating parts-feeding plans from part geometry. The algorithm can generate either squeeze plans or push–grasp plans for the class of polygonal parts. This planning algorithm has been implemented in Common Lisp using exact arithmetic to represent angles [43] and has been tested on hundreds of polygonal shapes with running times of well under 1 s on a Sun Sparcstation IPC.

The plans generated by this algorithm have been verified experimentally both at Carnegie Mellon University using a PUMA robot with an electric LORD Co. gripper and at the University of Southern California using an IBM robot with a pneumatic Robotics and Automation Corp. gripper. Using a pushing distance of two part diameters, the push–grasp plans for the house and 4-gon (shown in figures) performed as predicted in almost all trials. Occasionally, the part failed

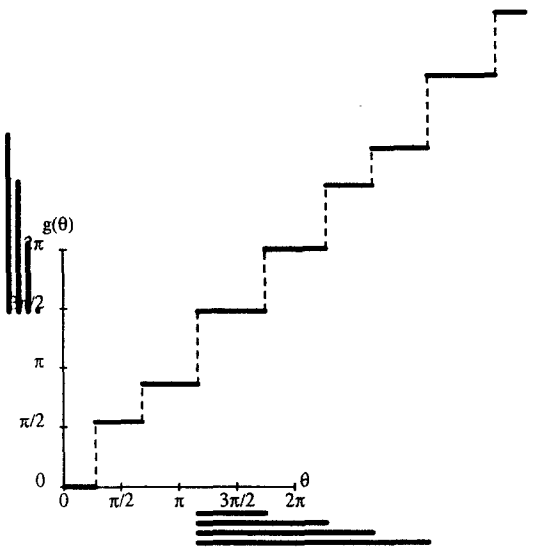

Fig. 11. The push–grasp function for the 4-gon (unwrapped to show two periods). The bars under the horizontal axis indicate the sequence of s-intervals found by the algorithm, the bars to the left of the vertical axis indicate the corresponding s-images.

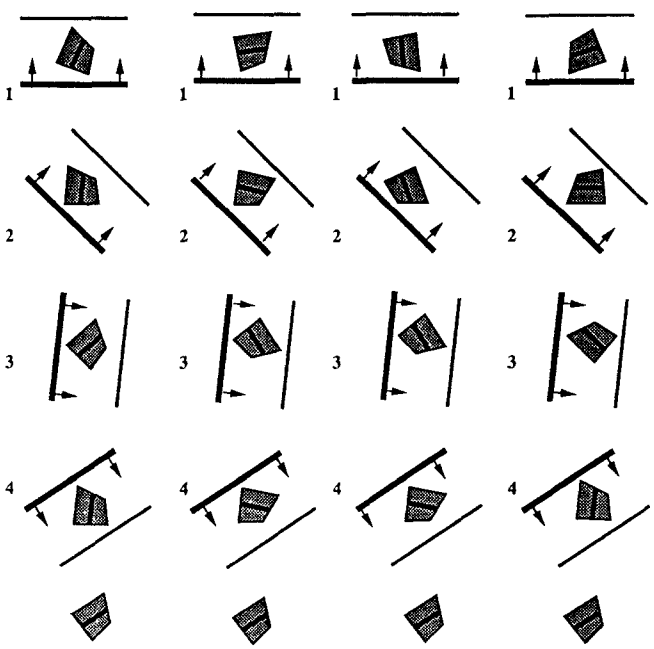

Fig. 12. Four traces of the push–grasp plan for the 4-gon, where the thick line indicates the pushing jaw. Push–grasp actions are at 0°, −44°, −96°, and −146°.

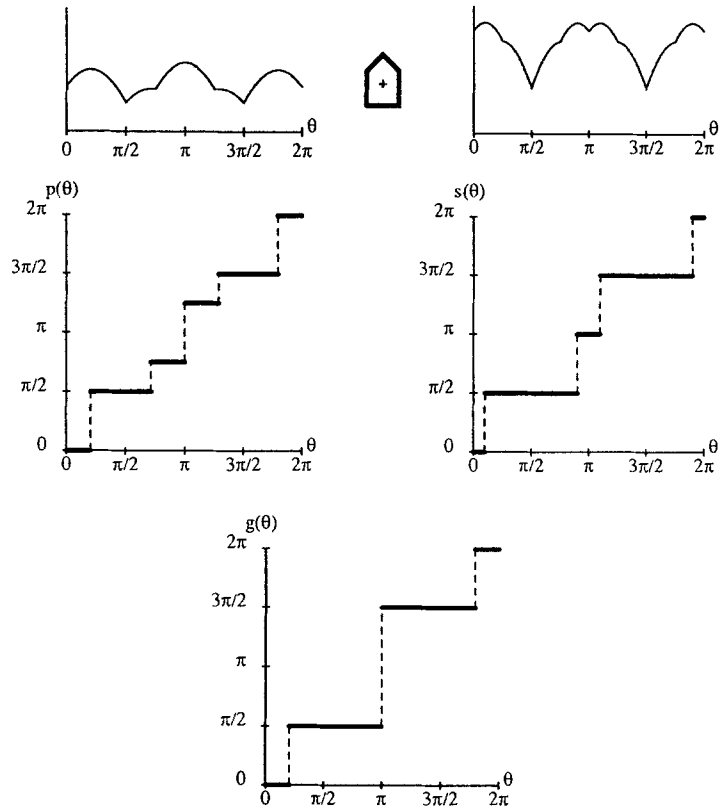

Fig. 13. Push–grasp analysis for the house-shaped part. The radius function is in the upper left. Directly below it is the push function. The diameter function is in the upper right. Directly below it is the squeeze function. The push–grasp function is shown at the bottom.

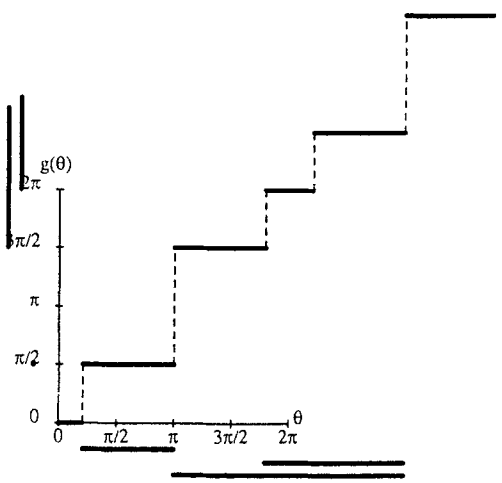

Fig. 14. The push–grasp function for the house-shaped part (unwrapped to show two periods). The bars under the horizontal axis indicate the sequence of s-intervals found by the algorithm, the bars to the left of the vertical axis indicate the corresponding s-images.

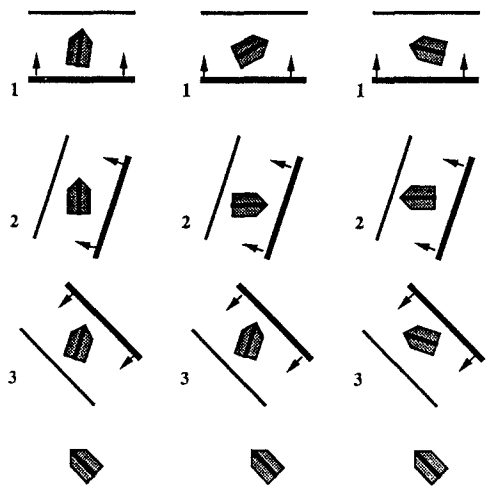

Fig. 15. Four traces of the push–grasp plan for the house-shaped part, where the thick line indicates the pushing jaw. Push–grasp actions are at 0°, 72°, and 135°.

to rotate into a stable configuration during the pushing phase. This may be due to insufficient pushing distance or error in our calculation of the part's center of mass arising from errors in estimating part geometry and/or nonuniform part density. Thus two important issues for future research are improved estimates for the minimum distance required to align a stable edge of the part with the pushing jaw and planning algorithms that are robust to variations in part shape and center of mass.

For this algorithm to be practical in an industrial environment, we must also consider the issue of feedrate. One idea is to pipeline the plans; rather than perform i actions with one gripper, use i fixed grippers so that parts flow out of the pipeline at the rate of one per action. Of course we must still be concerned with the length of the pipeline. We can make a tradeoff between plan length and the probability of success by optimizing *expected* feedrate, where the distribution of inputs is assured by randomizing the first gripper angle and a binary filter is used to reject parts that are not in the most probable orientation [18].

Furthermore, although we have shown that the plan length is bounded by $O(n^2)$, we conjectured based on empirical evidence that plan length is linear in the number of part edges. Recently, Yui-Bin Chen and Ierardi [20] proved that plan length is indeed linear which means that our algorithm will run in time $O(n^2)$ and produce plans of length $O(n)$. Also, Prasanna and Rao [44] reported a parallel version of this algorithm that runs in linear time on an $n \times n$ mesh-connected computer.

Although we assume that the part remains between the jaws throughout the plan, one degree of translational freedom is not constrained by the parts feeder. To resolve this after the part is oriented, we might allow the part to slide down a track until it falls through a silhouette trap, although it could be difficult to ensure that the part does not rotate during this phase. We are currently investigating an alternate approach based on open-loop pushing.

The method described here is limited to the class of planar polygonal parts: essentially flat parts with a constant polygonal cross-section. Many parts used in industry have curved edges and spatially varying cross-sections. For example, many computer-aided design systems and numerical control machines allow a cross-section to be defined with a combination of linear and piecewise-circular edges. We are currently extending the algorithm to cope with such shapes. For three-dimensional polyhedral parts, the planar algorithm might be preceded by a preliminary stage that causes a most stable face of the part to be aligned with the worktable. This might be accomplished using either vibration [24] or by tilting the worktable [35]. For some parts, we may be able to determine the remaining degree of rotational freedom using the algorithm described here to orient the polyhedron's two-dimensional projection.

In summary, we have described a planning algorithm that can rapidly analyze part geometry. The resulting plans require no sensors and can be pipelined to achieve rapid feedrate. The required hardware is widely available at low cost. Thus the resulting parts feeder is fully programmable and can be automatically programmed as part geometry changes.

By establishing that an algorithm is complete in the sense that it works for all parts in some class, the algorithm can be used as a dependable component in a modular automation system [45]. However, as discussed in Section 2, such algorithms are rare for compliant motion planning since we can usually construct cases for which a guaranteed plan does not exist. Making an analogy with the theory of compilers, which has defined classes of languages that can be automatically compiled, Natarajan [16] noted that "It would be useful to identify [a] class of parts for which feeder design can be easily automated." In this paper we have given a parts-orienting algorithm for the class of polygonal parts.

Acknowledgments. I am grateful to Matt Mason for advice and guidance throughout this research. Michael Erdmann suggested rereading [16] and was the first to suggest that a polynomial-time algorithm may exist. I am indebted to Randy Brost, John Canny, Merrick Furst, Doug Ierardi, Gary Miller, Barak Pearlmutter, Anil Rao, and the anonymous reviewers for constructive suggestions on improving an earlier draft. I also thank members of the CMU manipulation laboratory for their support: Srinivas Akella, Alan Christiansen, Kevin Lynch, and Yu Wang; and I would like to thank Klaus Gross and Rakkiat Trimahaloek for help in testing the resulting plans in the laboratory.

Appendix. The Diameter Function. In Section 3.1 we showed that a diameter function could be used to analyze the mechanics of squeeze–grasp actions. In this appendix we describe an $O(n)$ algorithm for computing the diameter function.

Let a two-dimensional part be described as a closed, compact region of E^2 which is also the closure of its interior [46]. For a fixed orientation of the part, define the part's *diameter* at direction θ to be the maximum distance between two parallel supporting lines at angle θ. Thus the **diameter function**, $d: S^1 \to \Re$,

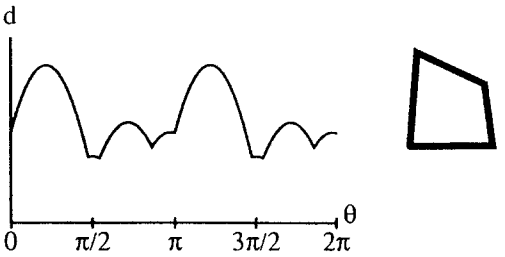

Fig. 16. The diameter function for the four-sided part shown at the right in its zero orientation. During a squeeze, the part rotates so as to reduce the diameter, terminating when the diameter reaches a local minimum.

describes how this distance varies as the parallel lines are rotated around the part (see Figure 16). Note that the diameter is simply the distance separating two jaws of the gripper when both jaws are just touching the part.

Jameson [47] used this function to show that any two-dimensional convex body must have at least two stable equilibria where it can be grasped between parallel jaws. In clustering applications, the maximum of the diameter function is known as the *diameter* of the set covered by the part [48]. The diameter function is also known as the *width function* in geometry,[7] but we use the former term to make an analogy with the radius function. We state the following without proof:

- The diameter function is continuous: $\Delta d \to 0$ as $\Delta\theta \to 0$.
- The diameter function for a part is equal to the diameter function for the convex hull of the part.
- The diameter function has period π (due to symmetry of the gripper) and may also have smaller period if the part's convex hull has rotational symmetry.

For a polygonal part whose convex hull is described by a list of n rational vertices, one period of the diameter function can be described by an ordered list of sinusoidal functions (phase and amplitude) and the associated transition angles. Transitions between sinusoids can only occur when an edge is aligned with the gripper, so there are at most $2n$ sinusoidal pieces. Since each sinusoid arises from contact between two opposing vertices in the object, we can trivially compute the diameter function by enumerating all n^2 pairs of vertices.

However, we can compute the diameter function in time $O(n)$ if we assume that the following operations require time $O(1)$: finding the distance between two points in the plane, finding the angle between two lines in the plane, comparing two rational numbers, adding or subtracting two rational numbers, and random access of any particular memory location. Preparata and Shamos [48] describe a linear-time algorithm for finding the maximum diameter of a convex polygon with n sides. Their algorithm enumerates all pairs of vertices that admit parallel supporting lines in order of increasing θ. Each pair defines a *chord* of length l_i and there are at most $3n/2$ such pairs. Thus to compute the diameter function, we

[7] For example, see the discussion of *curves of constant width* in [49].

make one pass over this list of chords as follows. Between every adjacent pair of part edges, $d(\theta)$ follows the sinusoid $l_i|\sin(\theta_i - \theta)|$, where $l_i$and θ_i are taken from the longest chord in the interval orthogonal to the interval between edges. A single sweep through the list of edges and chords allows us to compute the diameter function in linear time.

References

[1] J. H. Reif. Complexity of the mover's problem and generalizations. In *Proceedings of the 20th Symposium on the Foundations of Computer Science*, 1979.

[2] J. Schwartz, J. Hopcroft, and M. Sharir. *Planning, Geometry, and Complexity of Robot Motion.* Ablex, Norwood, NJ, 1987.

[3] J. F. Canny. The Complexity of Robot Motion Planning. Ph.D. thesis, MIT, May 1987.

[4] J.-C. Latombe. *Robot Motion Planning.* Kluwer Academic Press, Boston, 1991.

[5] T. Lozano-Perez, M. T. Mason, and R. H. Taylor. Automatic synthesis of fine-motion strategies for robots. *International Journal of Robotics Research*, 3(1):3–24, Spring 1984.

[6] M. A. Erdmann. On Motion Planning with Uncertainty. Master's thesis, MIT, August 1984.

[7] B. R. Donald. *Error Detection and Recovery in Robotics.* Springer-Verlag, New York, 1987.

[8] R. C. Brost. Analysis and Planning of Planar Manipulation Tasks. Ph.D. thesis, Carnegie Mellon University, January 1991.

[9] S. J. Buckley. Planning and Teaching Compliant Motion Strategies. Ph.D. thesis, MIT, January 1986.

[10] A. Briggs. An efficient algorithm for one-step compliant motion planning with uncertainty. In *Proceedings of the 5th Symposium on Computational Geometry*, 1989.

[11] J.-C. Latombe. Motion planning with uncertainty: On the preimage backchaining approach. In *The Robotics Review*, Vol. I, edited by O. Khatib, J. J. Craig, and T. Lozano-Perez. MIT Press, Cambridge, MA, 1989, pp. 53–70.

[12] B. R. Donald. Planning and executing robot assembly strategies in the presence of uncertainty. In *Proceedings of the 96th Annual Meeting of the American Mathematical Society*, 1990.

[13] S. Shekhar and J.-C. Latombe. On goal recognizability in motion planning with uncertainty. In *Proceedings of the International Conference on Robotics and Automation*, April 1991.

[14] J. Canny and J. Reif. New lower-bound techniques for robot motion planning problems. In *Proceedings of the 27th Symposium on the Foundations of Computer Sciencer*, 1987.

[15] T. Lozano-Perez. Motion planning and the design of orienting devices for vibratory part feeders. Unpublished memorandum, January 1986.

[16] B. K. Natarajan. Some paradigms for the automated design of parts feeders. *International Journal of Robotics Research*, 8(6):98–109, December 1989. Also appeared in *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, 1986.

[17] M. T. Mason. Kicking the sensing habit. In *Proceedings of the Asilomar Winter Workshop*, November 1991.

[18] K. Y. Goldberg. Stochastic Plans for Robotic Manipulation. Ph.D. thesis, School of Computer Science, Carnegie Mellon University, August 1990. Copies can be obtained from Catherine Copetas (copetas@cs.cmu.edu, 412-268-8525).

[19] M. A. Erdmann and M. T. Mason. An exploration of sensorless manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1986. Also appears in *IEEE Journal of Robotics and Automation*, 4(4), August 1988.

[20] Yui-Bin Chen and D. Ierardi. The Complexity of Oblivious Plans for Orienting Polygonal Parts. Technical Report USC-CS-92-502, University of Southern California, December 1991.

[21] G. Boothroyd, C. Poli, and L. E. Murch. *Automatic Assembly.* Marcel Dekker, New York, 1982.

[22] B.-Z. Sandler. *Robotics: Designing the Mechanisms for Automated Machinery.* Prentice-Hall, Englewood Cliffs, NJ, 1991. Chapter 7 summarizes results originally published (in Russian) in A. N. Rabinovitsh, *Automatic Orientation and Parts Feeding*, Technika, 1968.

[23] H. Hitakawa. Advanced parts orientation system has wide application. *Assembly Automation*, 8(3), 1988.

[24] N. C. Singer and W. Seering. Utilizing dynamic stability to orient parts. *Journal of Applied Mechanics*, 54, 1987. See also Singer's 1985 MIT Masters thesis.

[25] D. D. Grossman and M. W. Blasgen. Orienting mechanical parts by computer-controlled manipulator. *IEEE Transactions on Systems, Man, and Cybernetics*, 5, 1975.

[26] M. T. Mason. Manipulator Grasping and Pushing Operations. Ph.D. thesis, MIT. June 1982. Published in *Robot Hands and the Mechanics of Manipulation*, MIT Press, Cambridge, MA, 1985.

[27] V. T. Rajan, R. Burridge, and J. T. Schwartz. Dynamics of a rigid body in frictional contact with rigid walls. In *Proceedings of the IEEE Conference on Robotics and Automation*, 1987.

[28] R. C. Brost and M. T. Mason. Graphical analysis of planar rigid-body dynamics with multiple frictional contacts. In *Proceedings of the 5th International Symposium on Robotics Research*, 1989.

[29] R. C. Brost. Automatic grasp planning in the presence of uncertainty. *The International Journal of Robotics Research*, December 1988. Also appeared in *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, CA, April, 1986.

[30] M. Mani and R. D. W. Wilson. A programmable orienting system for flat parts. In *Proceedings of the North American Manufacturing Research Institute Conference XIII*, 1985.

[31] M. A. Peshkin and A. C. Sanderson. Planning robotic manipulation strategies for workpieces that slide. *IEEE Journal of Robotics and Automation*, 4(5), October 1988.

[32] K. Y. Goldberg and M. T. Mason. Bayesian grasping. In *Proceedings of the International Conference on Robotics and Automation*, May 1990.

[33] A. D. Christiansen. Automatic Acquisition of Task Theories for Robotic Manipulation. Ph.D. thesis, School of Computer Science, Carnegie Mellon University, 1991.

[34] D. Eppstein. Reset sequences for monotonic automata. *SIAM Journal of Computing*, 19(5), 1990.

[35] M. Erdmann, M. T. Mason, and G. Vanecek, Jr. Mechanical parts orienting: The case of a polyhedron on a table. In *Proceedings of the International Conference on Robotics and Automation*, April 1991.

[36] K. Y. Goldberg, M. T. Mason, and M. A. Erdmann. Generating stochastic plans for a programmable parts feeder. In *Proceedings of the International Conference on Robotics and Automation*, April 1991.

[37] M. T. Mason. On the scope of quasi-static pushing. In *Proceedings of the Third International Symposium on Robotics Research*, edited by O. Faugeras and G. Giralt, MIT Press, Cambridge, MA, 1986.

[38] M. A. Peshkin. Planning Robotic Manipulation Strategies for Sliding Objects. Ph.D. thesis, Department of Physics, Carnegie Mellon University, November 1986.

[39] R. H. Taylor, M. M. Mason, and K. Y. Goldberg. Sensor-based manipulation planning as a game with nature. In *Robotics Research: The Fourth International Symposium*. MIT Press, Cambridge, MA, 1988.

[40] M. T. Mason, K. Y. Goldberg, and R. H. Taylor. Planning Sequences of Squeeze–Grasps To Orient and Grasp Polygonal Objects. Technical Report CMU-CS-88-127, Computer Science Department, Carnegie Mellon University, April 1988.

[41] K. Goldberg. A kinematically-yielding gripper. In *Proceedings of the 22nd International Symposium on Industrial Automation*, October 1991. See U.S. Patent #5,098,145, granted March 1992.

[42] K. Y. Goldberg and A. Rao. Orienting Planar Parts. Technical Report IRIS-275, University of Southern California, September 1991.

[43] M. T. Mason. Exact angle arithmetic package in commonlisp. Private communication, August 1988.

[44] V. K. Prasanna and A. S. Rao. Parallel orientation of polygonal parts. Technical Report CEng-91-23, University of Southern California, August 1991.

[45] M. Erdmann. Personal communication, 1990.

[46] A. A. G. Requicha. Representations for rigid solids: Theory, models, and systems. *ACM Computing Surveys*, 12:437–464, 1980.

[47] J. Jameson. Analytic Techniques for Automated Grasp. Ph.D. thesis, Department of Mechanical Engineering, Stanford University, June 1985.

[48] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, 1985.

[49] I. M. Yaglom and V. G. Boltyanskii. *Convex Figures*. Holt, Rinehart and Winston, New York, 1961.