

Orienting Parts with Shape Variation

Fatemeh Panahi, Mansoor Davoodi and A. Frank van der Stappen

Abstract Industrial parts are manufactured to tolerances as no production process is capable of delivering perfectly identical parts. It is unacceptable that a plan for a manipulation task that was determined on the basis of a CAD model of a part fails on some manufactured instance of that part, and therefore it is crucial that the admitted shape variations are systematically taken into account during the planning of the task. We study the problem of orienting a part with given admitted shape variations by means of pushing with a single frictionless jaw. We use a very general model for admitted shape variations that only requires that any valid instance must contain a given convex polygon P_I while it must be contained in another convex polygon P_E . The problem that we solve is to determine, for a given h , the sequence of h push actions that puts all valid instances of a part with given shape variation into the smallest possible interval of final orientations. The resulting algorithm runs in $O(hn)$ time, where $n = |P_I| + |P_E|$.

Keywords Part feeding · Shape variation · Pushing

1 Introduction

Most of the existing solutions in algorithmic automation assume a severely idealized world in which parts are perfectly identical to their CAD-model and manipulators and sensors are infinitely accurate. In real life, however, parts are manufactured to

F. Panahi (✉) · A.F. van der Stappen
Department of Information and Computing Sciences, Utrecht University, P.O. Box 80089, 3508
TB Utrecht, The Netherlands
e-mail: f.panahi@uu.nl

A.F. van der Stappen
e-mail: A.F.vanderStappen@uu.nl

M. Davoodi
Department of Computer Science and Information Technology, Institute for Advanced Studies in
Basic Sciences (IASBS), P.O. Box 45137-66731, Zanjan, Iran
e-mail: mdmonfared@iasbs.ac.ir

tolerances [6, 7] and therefore vary in shape [1], and sensors [4] and actuators [5] are inaccurate, causing the aforementioned algorithms to fail when employed in practice. The challenge is therefore to design algorithms for planning manipulation tasks that explicitly take into account manipulator (and sensor) inaccuracy and part imperfection and report solutions that work despite their presence. In this paper, we concentrate on part shape variation and study its impact on the problem of feeding or orienting it by means of pushing with a frictionless jaw in the spirit of the work by Goldberg [11]. We employ a very general model for shape variation and systematically explore its impact on the task of orienting by pushing. We use the resulting properties to develop a robust algorithm that reduces the uncertainty in the pose of an *imperfect part*, i.e., a part with shape variation, as much as possible.

Sensorless manipulation has received considerable attention over the past two decades. It focuses on manipulation systems that use simple (and thus cheap and reliable) hardware components that are only capable of performing simple physical actions while using simple or no sensors. The goal in sensorless part feeding or orienting is to reduce the set of possible orientations until the part is in a known final orientation. Lozano-Perez et al. [8] and Erdmann and Mason [9] proposed designs for feeding based on a finite set of actions to orient a part. Akella and Mason [10] developed a complete open-loop plan for feeding by means of pushing. Goldberg [11] showed that there always exists a plan for orienting a polygonal part by pushing or squeezing using a frictionless parallel-jaw gripper and proposed a greedy algorithm for computing the shortest such plan in $O(n^2 \log n)$ time, where n is the number of vertices of the part. He conjectured that the length of the shortest plan is linear in n . Chen and Ierardi [12] proved Goldberg's conjecture and also showed how to compute the maximum uncertainty radius such that a plan still exists. Berretty et al. [13] showed that 3D (polyhedral) parts can be oriented by a sequence of pushes by a perpendicular pair of planar jaws and gave an $O(n^3 \log n)$ time algorithm to find such a plan.

There are also approaches that are based on constrained forms of pushing. Wiegley et al. [14] considered a system consisting of a conveyor belt with fences mounted to its sides, which reorient parts that slide along them while traveling on the belt. The problem of designing the fences is equivalent to computing push actions with constraints on successive push directions. Wiegley et al. presented an exponential algorithm for finding the shortest sequence of fences that orients a given part. Berretty et al. [15] presented an alternative graph-based algorithm that runs in $O(n^3 \log n)$ time. In addition, Goldberg [11] and Chen and Ierardi [12] also studied grasps in which a jaw first pushes and then squeezes a part. Their time and complexity bounds are similar to those for pure pushing.

Several authors considered models and problems involving uncertainty in geometric data. For modeling shape variation, geometric approaches such as ϵ -geometry [16], λ -geometry [17], tolerance and interval-geometry [18, 20] were proposed. In these models, imprecise input data (e.g. vertices of a polygon) are constrained to vary in a region such as a segment, disk, rectangle, or any convex polygon, and worst and best cases of the output of certain problems are studied.

There have been a few studies into part feeding in a context of imperfect parts. Akella and Mason [22] studied the problem of orienting convex polygons whose vertices and center of mass lie inside predefined disks centered at their nominal locations. They required that any variation keeps the part convex. They proposed graph-based approaches for fence and push-squeeze plans for parts that satisfy their assumptions. The problem of orienting a part by fences has been studied by Chen et al. [3]. They used a similar model for part shape variation by allowing the vertices to vary inside disks and squares that are defined relative to the center of mass. Based on their assumptions they proposed a method for computing the maximum allowable disk or square for each vertex for feeding. In addition to these studies, other related work [23, 24] considered location uncertainty and shape variation in a grasp planning context.

In comparison with the aforementioned studies, we consider a more general model for shape variation that allows to characterize variation along the entire boundary instead of only at the vertices. The model assumes that any valid instance of a part contains a given convex shape while it is contained in another given convex shape. Our goal is to solve the part feeding or orienting problem for the imperfect part, that is, we want to find the sequence of pushes that puts all instances from the shape family into the smallest possible interval of orientations. To this end we generalize the notions of radius and push function [11] to families of shapes. In Sects. 3 and 4 we present several properties of the generalized push function along with its upper and lower envelopes. These properties help us to develop a greedy algorithm for reporting the smallest interval of possible orientations for the entire shape family after a given number h of pushes. We also show that there exist imperfect parts for which there always is a next push that shrinks the interval of possible orientations.

2 Preliminaries

In this section, we explain our assumptions and introduce the terminology and notation used throughout the paper. To do this, we first define the problem of orienting a part with shape variation. Then, we will have a short review of the relevant concepts from previous work and finally we define similar concepts for a part with shape variation. For brevity, we have omitted most proofs in this paper, but interested readers can find those in our technical report [27].

2.1 *Orienting Parts with Shape Variation*

Manufactured parts always have slight imperfections; hence, they are designed up to certain tolerances. We study the problem of orienting a part with shape variations by means of pushing with a single frictionless jaw [11] under the general shape variation model presented in [19]. In this model, for any manufac-

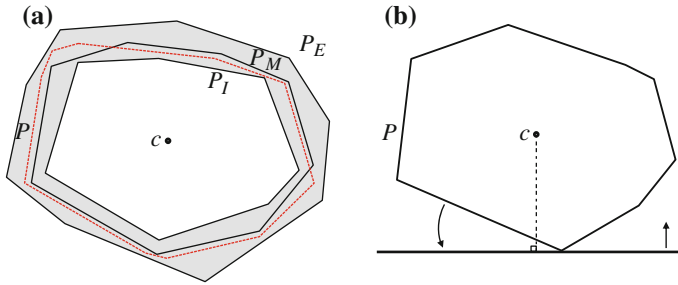


Fig. 1 **a** A family of shapes specified by a subshape P_I and a supershape P_E of a model part P_M along with a valid instance $P \in S(P_I, P_E)$. **b** A polygon P and its supporting line in the vertical downward direction; when the single jaw moves upward, P rotates in counterclockwise direction

tured planar model part of P_M , the set of acceptable instances is a family of shapes $S(P_I, P_E) = \{P \subset \mathbb{R}^2 \mid P_I \subseteq P \subseteq P_E\}$ where P_I and P_E are two given closed objects satisfying $P_I \subseteq P_M \subseteq P_E$. The closed region resulting from subtracting the interior of P_I from P_E is referred to be *tolerance zone* and denoted by Q . See Fig. 1a. We will often refer to a part with shape variation as an imperfect part. The objects P_I and P_E in this paper are assumed to be convex and polygonal with a total of n edges. The property of convexity helps us to compute a tight bound on the final orientation of an imperfect part. Also, we assume that the boundaries of P_I and P_E are disjoint.

When there is variation in part shape there will also be variation in the location of the center of mass of the part. In general, the problem of finding the exact locus of the center of mass for a polygon with shape variation has been mentioned as an open problem in [4, 22]. An algorithm for computing a polygonal approximation of the locus has been presented in [19] under the aforementioned shape variation model. However, for simplicity in this paper, we assume that all instances of an imperfect part have their center of mass at the origin. As a result, an instance P belongs to $S(P_I, P_E)$ if its boundary lies completely inside the tolerance zone Q when its center of mass is placed at the origin.

The basic action of pushing a part at the direction of θ consists of placing a single jaw in orientation θ and moving it in a direction perpendicular to itself. When a part P is pushed, it will start a compliant motion (rotation), during which it decreases the distance from its center of mass to the jaw. The motion stops when the normal to the jaw passes through the center of mass of the part. We refer to the corresponding direction of the contact normal as an *equilibrium orientation*. An equilibrium orientation is a *stable orientation* if an edge of part's convex hull is in contact with the jaw [2].

We define the problem of *orienting an imperfect part* to be that of finding the sequence of push actions that orients the part to the smallest possible orientation set. This possible orientation set consists of disjoint intervals. However, we do not exploit this fact and focus on finding the smallest single interval that contains all possible orientations.

2.2 Definitions for a Part

Throughout this paper, directions are relative to a fixed coordinate frame attached to the origin, increasing in counterclockwise order. Let the set of orientations of P be identified with points on the planar unit circle $S^1 : [0, 2\pi)$. For any orientation θ , the supporting line at the direction θ is a supporting line whose normal vector emanating from the origin has direction θ . See Fig. 1b. Pushing P at the direction θ means aligning the jaw with the supporting line at the direction θ . For an interval Θ , we let $L(\Theta)$ and $U(\Theta)$ be the lower and upper bounds (left and right endpoints) of Θ , respectively, and $|\Theta|$ be its length.

The radius function $r_P : S^1 \rightarrow \mathbb{R}^+$ of a part P maps an angle θ onto the distance between the center of mass and the supporting line of P at the direction θ [2]. The distance function $\delta_P : S^1 \rightarrow \mathbb{R}^+$ of P maps an angle θ onto the distance between the center of mass and the intersection point of the boundary ∂P of P and the ray emanating from the center of mass at the direction θ [13]. Figure 2 depicts the radius functions of P_I and P_E and the distance function of P_E for the illustrated imperfect part. The radius and distance functions are closely related; see Observation 1.

Observation 1 *The local minima and maxima of r_P and δ_P coincide; r_P is increasing (decreasing) if and only if δ_P is increasing (decreasing).*

The push function $\phi_P : S^1 \rightarrow S^1$ of P maps a push direction of the jaw relative to P in its reference orientation onto the orientation of P after alignment with the

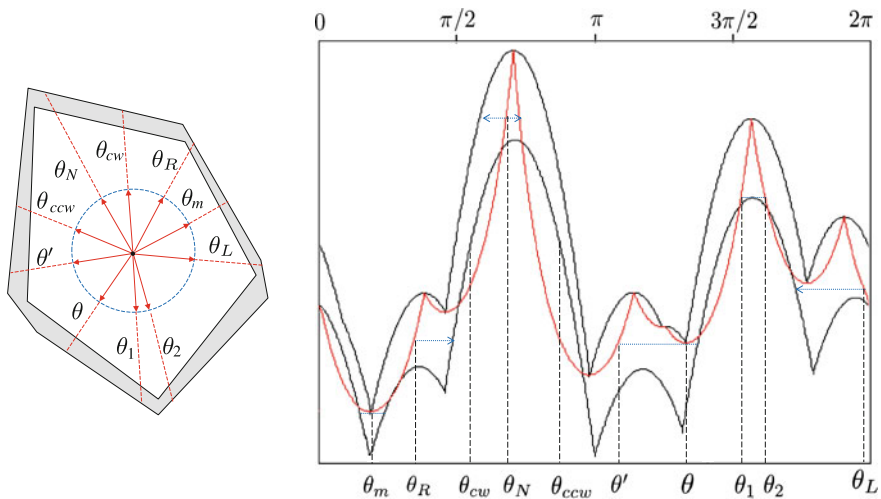


Fig. 2 An example of an imperfect part, the corresponding graphs of r_I, r_E are illustrated in black and the red graph depicts δ_E . The illustrated angles θ_R, θ_L and θ_N are R-type, L-type and N-type, respectively; θ_m is both R-type and L-type. θ_{cw} is a clockwise and θ_{ccw} is a counterclockwise unstable angle. $[\theta', \theta)$ and (θ_1, θ_2) are N-type intervals

jaw. It is well known [11] that the push function follows directly from the radius function as it maps all orientations that are strictly between two consecutive local maxima of the radius function onto the local minimum that is enclosed by these local maxima; moreover, the push function maps each local maximum of the radius function onto itself.

2.3 Definitions for a Part with Shape Variation

In this subsection, we define the relevant concepts related to imperfect parts. For simplicity, we use the abbreviations $r_I = r_{P_I}$, $r_E = r_{P_E}$, and $\delta_E = \delta_{P_E}$. Figure 2 illustrates an example of an imperfect part and the graph of r_I , r_E and δ_E . The following lemma shows that r_I and r_E bound the radius function of all instances of an imperfect part.

Lemma 1 $r_I \leq r_P \leq r_E$ for all $P \in S(P_I, P_E)$.

Pushing an imperfect part means pushing an unknown instance from a shape family $S(P_I, P_E)$. As a consequence, the outcome of such a push is the set of all orientations that might result after pushing any shape $P \in S(P_I, P_E)$. To capture this behavior we define the *generalized push function* $\Phi^* : S^1 \rightarrow \mathcal{P}(S^1)$, where $\mathcal{P}(S^1)$ denotes the power set of S^1 . This function maps an angle θ onto the set of all possible orientations after a single push action in the direction θ , so $\Phi^*(\theta) = \{\phi_P(\theta) | P \in S(P_I, P_E)\}$. As there are several ways to enclose the sets $\Phi^*(\theta)$ by a single interval (due to the cyclic nature of S^1) we must be careful when defining these intervals to avoid ambiguity. To this end we introduce the *lower push function* and the *upper push function* in Definition 1.

Definition 1 The *lower push function* $\Phi_L^* : S^1 \rightarrow S^1$ and *upper push function* $\Phi_U^* : S^1 \rightarrow S^1$ are the functions that bound Φ^* as follows. We consider three cases based on the push direction θ .

- (a) If all instances of $S(P_I, P_E)$ rotate clockwise when pushed at θ then let α and β be tight upper and lower bounds on the magnitude of the clockwise rotations, respectively. Then $\Phi_L^*(\theta) = \theta - \alpha$ and $\Phi_U^*(\theta) = \theta - \beta$.
- (b) If all instances of $S(P_I, P_E)$ rotate counterclockwise when pushed at θ then let α and β be tight lower and upper bounds on the magnitude of the counterclockwise rotations, respectively. Then $\Phi_L^*(\theta) = \theta + \alpha$ and $\Phi_U^*(\theta) = \theta + \beta$.
- (c) Otherwise let α and β be tight upper bounds on the magnitudes of the clockwise and counterclockwise rotations, respectively. Then $\Phi_L^*(\theta) = \theta - \alpha$ and $\Phi_U^*(\theta) = \theta + \beta$.

Note that for each $\theta \in S^1$ the interval $[\Phi_L^*(\theta), \Phi_U^*(\theta)]$ contains the set $\Phi^*(\theta)$. We will occasionally denote this interval by $\Phi(\theta)$ and refer to it as the smallest

interval containing the set $\Phi^*(\theta)$. Moreover, for an interval $\Theta \subseteq S^1$ we let $\Phi(\Theta) = [\Phi_L^*(L(\Theta)), \Phi_U^*(U(\Theta))]$

We also note that Φ_L^* and Φ_U^* are monotone (non-decreasing), which admits a greedy approach to orient the imperfect part into the smallest possible range of angles. We start with the initial set of possible orientations $\Theta_0 = [0, 2\pi)$ and repeatedly obtain Θ_{i+1} by selecting it to be the shortest image of any translate of Θ_i under Φ . The process continues as long as $|\Theta_{i+1}| < |\Theta_i|$. To this end, we need to compute the functions Φ_L^* and Φ_U^* . For different types of orientations, the values of these functions are computed differently. These types of angles are defined in the next section.

Remark Since range and domain of Φ_L^* and Φ_U^* are S^1 , it is possible that $\Phi_L^*(L(\Theta)) > \Phi_U^*(U(\Theta))$. In this case, $|\Phi(\Theta)| = 2\pi + \Phi_U^*(U(\Theta)) - \Phi_L^*(L(\Theta))$.

3 Types of Orientations

The set of all orientations can be divided into five types based on the computation of their image under Φ_L^* and Φ_U^* . We distinguish two primary types which consist of two and three subtypes respectively.

- An orientation θ is *unstable* if there is no $P \in S(P_I, P_E)$ for which r_P has a local minimum at θ . Such an orientation can never be the final orientation of the imperfect part after pushing. Unstable orientations can be (i) *clockwise unstable*, or (ii) *counterclockwise unstable*.
- An orientation θ is *potentially stable* or *p-stable* if there exists an instance $P \in S(P_I, P_E)$ for which r_P has a local minimum at θ . Such an orientation can be a final orientation of the imperfect part after pushing. Potentially-stable orientations can be (i) *right type (R-type)*, or (ii) *left type (L-type)*, or (iii) *neutral type (N-type)*.

In the following subsections we define the subtypes and properties of p-stable and unstable orientations. The types of orientations divide S^1 into intervals of orientation of the same type. These intervals will be referred to as *critical intervals*. The type of a critical interval equals the type of orientations it contains

3.1 Unstable Intervals

Unstable intervals help to reduce the uncertainty in the orientation of an imperfect part as they can never appear in the set of possible orientations after a push action. The following lemma describes how we can distinguish unstable angles.

Lemma 2 *An orientation $\theta \in S^1$ is unstable if and only if $\delta_E(\theta) < r_I(\theta)$.*

Figure 2 shows several unstable intervals, in which the (red) graph of δ_E lies below the (lower black) graph of r_I . Lemma 2 shows that we can determine the subdivision of S^1 into unstable and p-stable intervals by computing the intersection of δ_E and r_I .

Note that the unstable intervals can be computed in $O(n)$ time since the number of intersection points cannot exceed $O(n)$.

Observation 2 *Let $\Theta \subset S^1$ be an unstable interval. All instances $P \in S(P_I, P_E)$ will rotate in the same direction, i.e., in either clockwise or counterclockwise direction, for all push directions $\theta \in \Theta$.*

The above observation shows that there are clockwise and counterclockwise orientations and intervals. For any instance $P \in S(P_I, P_E)$, r_P is strictly increasing in a clockwise unstable interval and strictly decreasing in a counterclockwise unstable interval. In Fig. 2, θ_{cw} and its containing interval are clockwise unstable while θ_{ccw} and its containing interval are counterclockwise unstable.

3.2 Potentially-Stable Intervals

According to Lemma 2 p-stable orientations are angles in which the graph of δ_E lies above the graph of r_I . Now consider the graph of δ_E . A p-stable angle θ is called *R-type* if from the point $(\theta, \delta_E(\theta))$ the graph of r_I is horizontally visible to the right. Similarly, it is called *L-type* if the graph of r_I is horizontally visible to the left. If there is no horizontal visibility of r_I the p-stable angle is referred to as *N-type*. In Fig. 2 the angle θ_R is R-type because the horizontal ray emanating from $(\theta, \delta_E(\theta))$ to the right first hits r_I ; θ_L is an L-type angle as the horizontal ray emanating from $(\theta, \delta_E(\theta))$ to the left hits r_I .

The following definition describes the three types of angles more precisely.

Definition 2 Let $\theta \in S^1$ be a p-stable angle.

- θ is *R-type* if and only if there is no angle ξ such that $\theta < \xi < \theta'$ and $r_E(\xi) = \delta_E(\theta)$, where $\theta' > \theta$ is the smallest angle such that $r_I(\theta') = \delta_E(\theta)$. The angle θ' is called *upper bound* of θ denoted by $B_U(\theta)$.
- θ is *L-type* if and only if there is no angle ξ such that $\theta' < \xi < \theta$ and $r_E(\xi) = \delta_E(\theta)$, where $\theta' < \theta$ is the largest angle such that $r_I(\theta') = \delta_E(\theta)$. The angle θ' is called *lower bound* of θ denoted by $B_L(\theta)$.
- θ is *N-type* if it is neither R-type nor L-type.

Remark Throughout this paper, the term right refers to the counterclockwise direction and left refers to the clockwise direction. The following observation can be made about r_I and r_E . See Fig. 2.

Observation 3 *Let $\theta \in S^1$ be R-type (L-type). Then r_E is increasing (decreasing) in a sufficiently small right (left) neighborhood of θ and r_I is increasing (decreasing) in a sufficiently small left (right) neighborhood of $B_U(\theta)$ ($B_L(\theta)$).*

It is possible that an angle is both L-type and R-type. Lemma 3 shows that such angles are local minima of r_E .

Lemma 3 *If $\theta \in S^1$ is R-type and L-type, then θ is a local minimum of r_E .*

It is not difficult to see that each orientation is of one of the aforementioned types. Lemma 4 bounds the resulting number of critical intervals and their computation time.

Lemma 4 *There are $O(n)$ critical intervals; they are computable in $O(n)$ time.*

4 Computing the Lower and Upper Push Functions

To compute Φ_L^* and Φ_U^* , we need to find the tight lower and upper bounds for the amount of clockwise or counterclockwise rotation of an imperfect part. (See Definition 1.) Recall that when a part is pushed, it rotates in the direction in which the radius function decreases. As a result, we are interested in as longest as possible non-increasing curve (to the right as well as to the left) that lies completely between r_I and r_E . We note that not every such a curve corresponds to a valid part. Therefore, our strategy is to construct valid instances which create these bounds for clockwise and counterclockwise rotations when it is being pushed at θ .

In this section, we show that if θ belongs to a counterclockwise unstable interval then $\Phi_L^*(\theta)$ is the right endpoint of that interval. Otherwise, $\Phi_L^*(\theta)$ is the left bound of some specific L-type angle. Similarly, if θ belongs to a clockwise unstable interval then $\Phi_U^*(\theta)$ is the left endpoint of that interval. Otherwise, $\Phi_U^*(\theta)$ is the right bound of some specific R-type angle. We will focus on computing upper bounds in this section with the understanding that lower bounds can be computed similarly.

If θ is a clockwise unstable angle then there is no instance $P \in S(P_I, P_E)$ that rotates counterclockwise. Therefore, the upper bound cannot exceed the left endpoint of the unstable interval that contains θ . This upper bound is easy to compute. We now assume that θ is not a clockwise unstable angle. In this case, $\Phi_U^*(\theta) \geq \theta$. We note that if an instance P rotates counterclockwise, then r_P has to be strictly decreasing in a sufficiently small right neighborhood of θ . We define an instance whose radius function is decreasing along the largest possible interval. We refer to this instance as the *upper critical instance* at the direction θ . The critical instance suggests us an approach to compute $\Phi_U^*(\theta)$. We present an algorithm that constructs the upper critical instance for every θ . Then, we prove a theorem that helps to compute Φ_U^* from these critical instances.

By definition, if P is an upper critical instance, then r_P has to be decreasing in the interval $[\theta, \Phi_U^*(\theta)]$. For angles in which r_E is decreasing, it is not difficult to find such instances. For the other angles we prove the following lemma.

Lemma 5 *Let $\theta \in S^1$ be an angle such that r_E is increasing in a right neighborhood of θ and let $P \in S(P_I, P_E)$ be an instance that rotates counterclockwise after a single push action at the direction θ . Then $r_P(\theta) \leq \delta_E(\theta)$.*

The next corollary follows from Lemma 5 and Observation 1.

Corollary 1 *Let $\theta \in S^1$ be an R-type angle and $P \in S(P_I, P_E)$ be its upper critical instance. Then $r_P(\theta) \leq \delta_E(\theta)$.*

Corollary 1 reveals that $B_U(\theta)$ is an upper bound on $\Phi_U^*(\theta)$. Note that by Observation 3 for an R-type angle θ , r_I is ascending in the left neighborhood of $B_U(\theta)$. So, no decreasing curve starting in the right neighborhood of θ cannot extend beyond $B_U(\theta)$. The following lemma shows that $B_U(\theta)$ is tight.

Lemma 6 *Let $d > 0$ be a constant and $[\theta_1, \theta_2] \subset S^1$ be an interval such that for all $\theta \in [\theta_1, \theta_2]$, $r_I(\theta) \leq d \leq \delta_E(\theta)$. Then there is an instance P such that $r_P(\theta) = d$ for all $\theta \in [\theta_1, \theta_2]$.*

So far, we discussed how to compute $\Phi_U^*(\theta)$ if θ is clockwise unstable or R-type. Otherwise, we claim that there is an instance $P \in S(P_I, P_E)$ such that r_P is decreasing in $[\theta, B_U(\theta_m)]$, where θ_m is the closest R-type angle to θ in counterclockwise direction. If such an angle does not exist, then the upper bound is 2π . The following lemma shows that θ_m is a local minimum of r_E .

Lemma 7 *If an angle θ is neither a clockwise unstable angle nor an R-type angle, then the closest R-type angle to θ in counterclockwise direction is a local minimum for r_E .*

Algorithm 1 creates the upper critical instance for an angle θ_0 that is not clockwise unstable. The key idea is that for such an angle θ_0 , there is an instance $P \in S(P_I, P_E)$ such that r_P is decreasing in $[\theta_0, B_U(\theta_m)]$ where θ_m is the closest R-type angle to θ_0 in counterclockwise direction. If there is no such R-type angle, then there is an instance that can rotate arbitrarily close to 2π . We explain how to construct a decreasing function and then show that this function is a part of the radius function of the instance reported by Algorithm 1. Lemma 6 shows that any horizontal ray that lies above the graph of r_I and below the graph of δ_E lies on the radius function of some instance. Note that according to Lemma 5 for any $\theta \in [\theta_0, B_U(\theta_m)]$ if r_E is increasing in the neighborhood of θ and P rotates in counterclockwise direction, then $r_P(\theta) < \delta_E(\theta)$. Therefore, we construct a function for P by starting from θ_0 and follow the horizontal ray emanating from $(\theta_0, \delta_E(\theta_0))$ as long as it stays below δ_E and above r_I . Here P satisfies $r_P(\theta) = \delta_E(\theta_0)$. If the ray hits r_I we are done. Alternatively, it hits δ_E at some angle θ' at which δ_E is decreasing in the right neighborhoods of θ' . We continue by choosing $r_P(\theta) = \delta_E(\theta') \cos(\theta' - \theta)$ until we hit r_E . Then we follow r_E until the closest local minimum and then again we use horizontal rays and continue similarly. The blue graph in Fig. 3 is an example of a function that is created using this procedure. Algorithm 1 constructs the corresponding instance which is also shown in Fig. 3. In Algorithm 1, $P(\theta_1, \theta_2)$ stands for the part of P between two rays emanating from the center of mass in directions θ_1 and θ_2 , E_I and E_E are the sets of edges of P_I and P_E respectively, and D_d is the boundary of a disc of radius d centered at the center of mass.

Algorithm 1 Constructing the upper critical instance

```

1: procedure CONSTRUCT  $Q(\theta_0)$  ▷  $\theta_0$  is not a cw unstable angle
2:    $Q \leftarrow null$  ▷ Initialization
3:    $Continue \leftarrow True$ 
4:    $d \leftarrow \delta_E(\theta_0)$ 
5:   if  $\partial D_d$  lies inside the tolerance zone then
6:      $Q \leftarrow D_d$  ▷ Upper critical instance is a disc
7:   else
8:     while ( $Continue$ )
9:        $d \leftarrow \delta_E(\theta_0)$ 
10:       $\theta_1 \leftarrow$  The closest angle to  $\theta_0$  in ccw direction such that  $\partial D_d$  intersects
11:        the segment  $s \in E_I \cup E_E$  at the direction  $\theta_1$  and  $\theta_0 \neq \theta_1$ 
12:      if  $s \in E_I$  then ▷  $\theta_0$  is an R-type angle
13:         $e \leftarrow$  the segment on the tangent line of  $D_d$  and  $P_I$  between them.
14:         $\alpha \leftarrow$  The direction of the normal vector of  $e$ .
15:         $Q \leftarrow Q \cup D_d(\theta_0, \alpha) \cup e$ 
16:         $Continue \leftarrow False$ 
17:      else
18:        if  $D_d(\theta_0, \theta_1)$  is inside the tolerance zone then
19:           $Q \leftarrow Q \cup D_d(\theta_0, \theta_1)$ 
20:           $\theta_0 \leftarrow \theta_1$ 
21:        else
22:           $\theta_m \leftarrow$  the closest local minimum of  $r_E$  to  $\theta_0$  in ccw direction.
23:           $Q \leftarrow Q \cup P_E(\theta_0, \theta_m)$ 
24:           $\theta_0 = \theta_m$ 
25:      Construct the rest of  $P$  arbitrarily to make it a valid instance.
26: end procedure

```

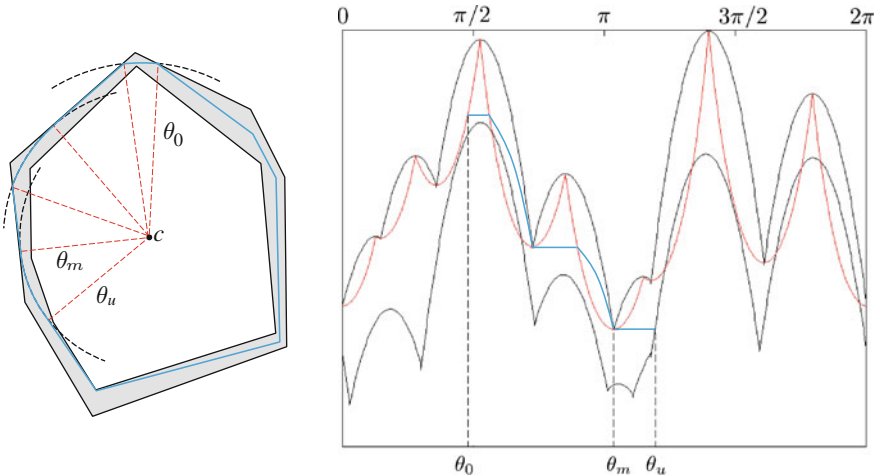


Fig. 3 Illustration of Algorithm 1 for an imperfect part. The critical instance constructed for the given angle θ_0 is shown in blue. The diagram on the right shows that the corresponding radius function is decreasing; θ_m is the closest R-type angle in counterclockwise direction from θ_0 and $\theta_u = B_U(\theta_m)$

The following lemmas provide the basis for the computation of the critical instance.

Lemma 8 *Let $\theta \in S^1$ be R-type and satisfying $\theta = B_U(\theta)$. There is no instance $P \in S(P_I, P_E)$ that rotates counterclockwise when pushed at θ .*

Lemma 9 *Assume that an imperfect part is pushed at direction θ_0 . If there is no $\theta \neq \theta_0$ such that θ is an R-type angle, then there is an instance in $S(P_I, P_E)$ that rotates arbitrarily close to 2π . Otherwise, we consider the following cases for the upper bound of the final orientation considering all instances $P \in S(P_I, P_E)$.*

- (a) *If θ_0 is a clockwise unstable angle, then the left endpoint θ_u of the containing unstable interval is a tight closed upper bound.*
- (b) *If θ_0 is not a clockwise unstable angle, then $\theta_u = B_U(\theta_m)$, with θ_m being the closest R-type angle to θ_0 in counterclockwise direction, is a tight open upper bound.*

We summarize the discussion of this section in the following theorem.

Theorem 4 Φ_L^* and Φ_U^* can be computed in $O(n)$.

Proof Lemma 4 shows that the critical intervals can be computed in $O(n)$. For any θ belonging to a critical interval Θ , the function Φ_U^* can be computed by applying Lemma 9.

- If Θ is **clockwise unstable**, then $\Phi_U^*(\theta) = L(\Theta)$.
- If Θ is **R-type**, then $\Phi_U^*(\theta) = B_U(\theta)$. Note that for an R-type angle θ , $r_I(\Phi_U^*(\theta)) = \delta_E(\theta)$. Then, $\Phi_U^*(\theta) = r_I^{-1}(\delta_E(\theta))$ for the corresponding range of δ_E and domain of r_I^{-1} .
- For all remaining intervals, i.e., **counterclockwise unstable, L-type and N-type**, $\Phi_U^*(\theta) = B_U(\theta_m)$ where θ_m is the closest R-type angle in counterclockwise direction. According to Lemma 7, θ_m is a local minimum of r_E ; therefore it is sufficient to check only the local minima. Since the number of local minima is linear and they occur in order, using a simple traversal of the graphs all of them can be computed in a linear time.

The time complexity of computing Φ_U^* is $O(n)$. The same bound applies to Φ_L^* . \square

Figure 4a, b illustrate an imperfect part and the radius and distance functions. The corresponding Φ_L^* and Φ_U^* are depicted by blue and black curves in Fig. 4c, respectively. It can be observed that for unstable and N-type intervals, the graphs of Φ_L^* and Φ_U^* are horizontal. For L-type intervals, the graph of Φ_L^* curves downward and the graph of Φ_U^* is horizontal, while for R-type intervals the graph of Φ_U^* curves upward and the graph of Φ_L^* is horizontal.

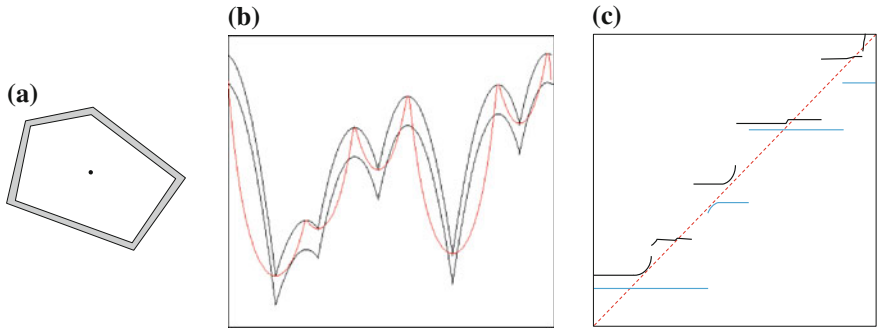


Fig. 4 **a** An example of an imperfect part. **b** r_L , r_E (black) and δ_E (red). **c** Φ_L^* (blue) and Φ_U^* (black)

5 An Algorithm for Orienting an Imperfect Part

In the previous section we have shown how to compute Φ_L^* and Φ_U^* . The monotonicity of these functions admits a greedy approach to find, for a given integer $h \geq 0$, the sequence of h push actions that orients an imperfect part into the smallest possible interval of orientations. Let Θ_i be the smallest interval containing all possible orientations after i pushes. Obviously, $\Theta_0 = S^1$, and after the first push the part will be in one of the p-stable orientations, so $\Theta_1 = S^1 - \Pi_{max}$, where Π_{max} is the largest unstable interval. The interval Θ_{i+1} can be obtained by computing the shortest image of any translate of Θ_i under Φ . The process continues as long as $|\Theta_{i+1}| < |\Theta_i|$ and $i < h$. Lemma 10 helps us to discretize the search for Θ_{i+1} by showing that it suffices to consider only translates of Θ_i in which one of its endpoints coincides with an endpoint of some unstable interval. We first give an observation that is needed to prove Lemma 10. It says that any p-stable angle θ appears in its own image under Φ^* (and Φ), because, by definition, there is an instance in $S(P_I, P_E)$ that is stable after pushing at θ .

Observation 5 $\Phi_L^*(\theta) \leq \theta \leq \Phi_U^*(\theta)$ if and only if θ is a p-stable angle, for any $\theta \in S^1$.

Lemma 10 Let $\Theta \subset S^1$ be an interval with the smallest image under Φ among all the intervals with the length of a given value. If $|\Phi(\Theta)| < |\Theta|$, then there exists an interval $\Theta' \subset S^1$ with $|\Phi(\Theta)| = |\Phi(\Theta')|$ such that $L(\Theta')$ or $U(\Theta')$ coincides with an endpoint of an unstable interval.

Algorithm 2 computes the smallest possible interval of orientations for an imperfect part after (at most) h push actions. Lemma 10 shows that it suffices to repeatedly align the endpoints of the current smallest interval Θ_i with each of the endpoints of the k unstable intervals Π_j ($1 \leq j \leq k$) to determine Θ_{i+1} . Figure 5 shows the application of the algorithm to the imperfect part of Fig. 4.

Algorithm 2 Compute the smallest possible orientation set

```

1: procedure COMPUTE-THE-SMALLEST-INTERVAL( $\Phi_L^*, \Phi_U^*, \Pi = \{\Pi_1, \Pi_2, \dots, \Pi_k\}, h$ )
2:    $i \leftarrow 1, X_1 \leftarrow 2\pi - \max_{1 \leq j \leq k} \{|\Pi_j|\}$  ▷ Initialization
3:    $Continue \leftarrow True$ 
4:   while ( $Continue$ ) and  $i \leq h$ 
5:      $S \leftarrow \emptyset$ 
6:     for  $j = 1$  to  $k$  ▷ For all unstable intervals
7:        $S \leftarrow S \cup \{[L(\Pi_j), L(\Pi_j) + X_i]\} \cup \{[U(\Pi_j) - X_i, U(\Pi_j)]\}$ 
8:        $\Theta_j \leftarrow \Theta \in S$  such that  $\forall \Theta' \in S, |\Phi(\Theta)| \leq |\Phi(\Theta')|$ 
9:       if ( $|\Phi(\Theta_j)| < X_i$ )
10:         $i \leftarrow i + 1$ 
11:         $X_i \leftarrow |\Phi(\Theta_j)|$ 
12:       else
13:         $Continue \leftarrow False$ 
14:   return  $\Theta_i, X_i$  for all  $1 \leq i \leq h$ 
15: end procedure

```

Theorem 6 *Algorithm 2 finds the sequence of $h \geq 0$ push actions that puts the imperfect part given by $S(P_I, P_E)$ in the smallest interval of possible orientations in $O(hn)$ time.*

Instead of running Algorithm 2 for a given maximum number h of pushes, we can also remove that bound and run it as long as the intervals Θ_i continue to shrink, to obtain the largest possible reduction of the uncertainty in the imperfect part’s pose. A natural question that arises is whether the algorithm would terminate in that case and thus whether the maximum reduction of pose uncertainty can be obtained after a finite number of pushes. It turns out that it is not always the case.

Recall that Algorithm 2 repeatedly aligns the left or right endpoint of an interval Θ_i with one of the $O(n)$ endpoints of an unstable interval Γ . The other endpoint of Θ_i then ends up in one of the $O(n)$ critical intervals, say Γ' . In order to obtain $\Omega(n^2)$ iterations the endpoints of some interval Θ_j with $j > i$ should be able to return to the same pair of intervals consisting of Γ and Γ' .

We assume without loss of generality that the left endpoint of Θ_i (and the future interval Θ_j) coincides with an endpoint of an unstable interval endpoint Γ_L . It is not hard to see that the interval Γ_R containing the right endpoint(s) must have a variable Φ_U^* .

Lemma 11 *Assume that the left endpoints of two intervals Θ_i and Θ_j in Algorithm 2 for some $j > i$ share the same endpoint of an unstable interval Γ_L and their right endpoints lie in the same critical interval Γ_R . Then Γ_R must be R-type.*

Lemma 12 helps us to determine the conditions for which the resulting intervals of Algorithm 2 can be shrunk endlessly.

Lemma 12 *Let $d > 0$ be a constant value and f be a continuous and non-decreasing function which has a derivative at every point in its domain $A \in \mathbb{R}$. Consider the*

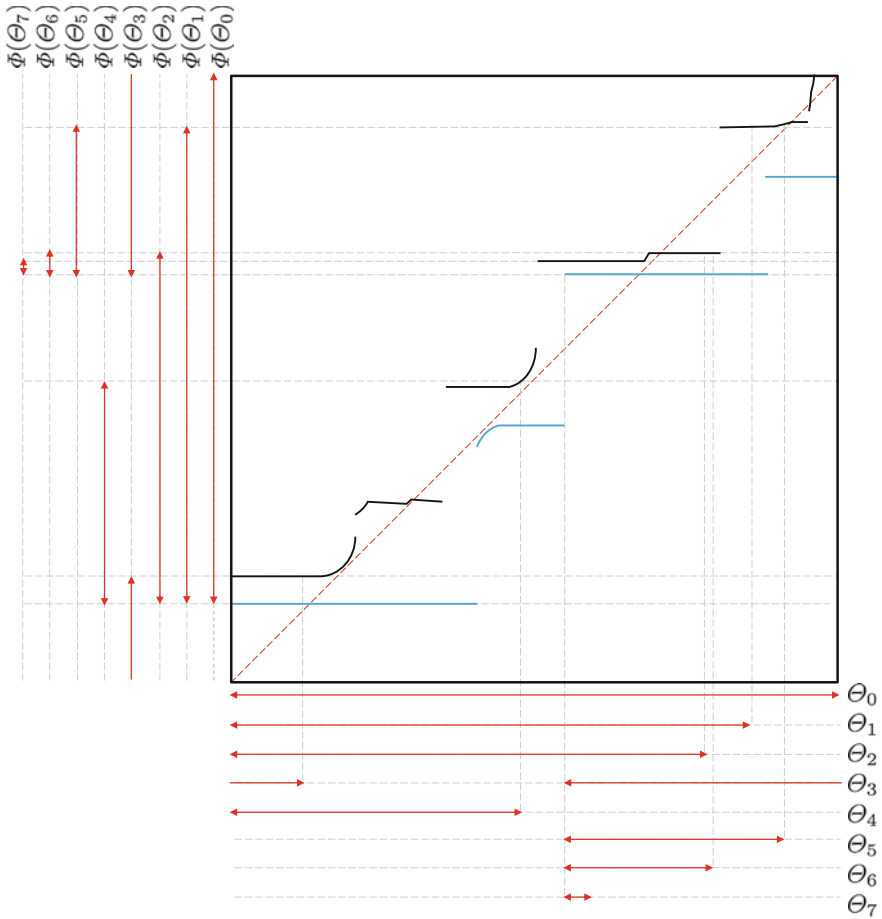


Fig. 5 Illustration of Algorithm 2 applied to the imperfect part of Fig. 4a, showing the intervals Θ_i for $i = 0, \dots, 7$. The length of the image of any translate of Θ_7 will be at least as long as Θ_7 ; as a result, no further reduction of the interval of possible orientations is possible

recursive sequence with the general term $x_{n+1} = f(x_n) - d$ and the first element $x_0 > 0$. If this sequence is decreasing and converges to some limit $a \in A$ then

- $f(x) < x + d, df/dx > 1$ where $x \in [a, x_0]$
- $f(a) = a + d$

Let Ψ_i ($i > 0$) be the i th interval that has its left endpoint in the unstable interval Γ_L and its right endpoint in the critical interval Γ_R . Assume that $i > 1$ and $|\Phi(\Psi_i)| < |\Psi_i|$. According to Observation 5, an unstable interval does not appear in its image while an R-type interval does appear in its image. See Fig. 6b. Considering the identity function and the image of Ψ_{i-1} , it can be observed that

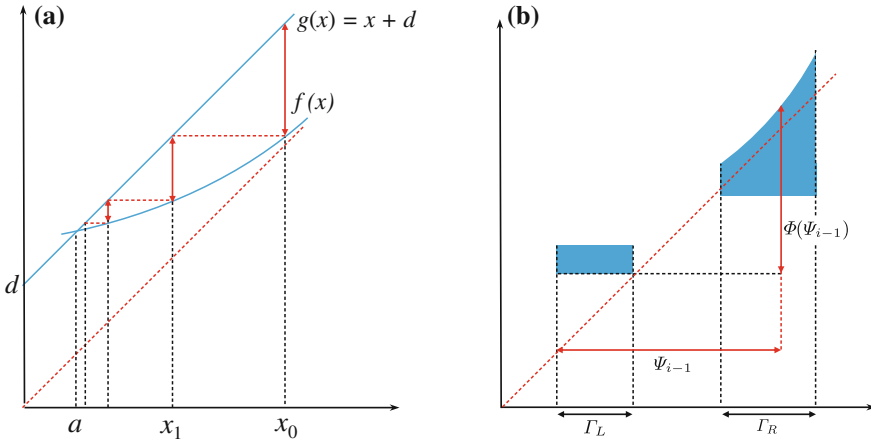


Fig. 6 **a** The sequence of $x_{n+1} = f(x_n) - d$ where $\lim_{n \rightarrow \infty} x_n = a$. **b** $L(\Psi_{i-1})$ lies on $L(\Gamma_L)$ and $U(\Psi_{i-1})$ lies in Γ_R ; $|\Phi(\Psi_{i-1})| = |\Psi_{i-1}| - |\Gamma_L| + \Phi_U^*(U(\Psi_{i-1})) - U(\Psi_{i-1})$

$|\Psi_i| \leq |\Phi(\Psi_{i-1})| = \Phi_U^*(U(\Psi_{i-1})) - \Phi_L^*(L(\Psi_{i-1})) = |\Psi_{i-1}| - |\Gamma_L| + \Psi(U(\Psi_{i-1}))$ where $\Psi(\theta) = \Phi_U^*(\theta) - \theta$. Therefore, $|\Psi_i| \leq \Phi_U^*(U(\Psi_{i-1})) - (|\Gamma_L| - L(\Psi_{i-1}))$. Note that $|\Gamma_L| - L(\Psi_{i-1})$ is a constant. According to Lemma 12 the smallest possible final orientation set can be obtained after a finite number of iterations unless the following conditions are met.

1. For $\theta \in \Gamma_R$ the graph of $f(\theta) = \theta + (|\Gamma_L| - L(\Psi_{i-1}))$ lies above the graph of Φ_U^* and $d\Phi_U^*/d\theta < 1$.
2. The graph of $f(\theta) = \theta + (|\Gamma_L| - L(\Psi_{i-1}))$ intersects Φ_U^* in Γ_R .

According to Lemma 12, if Ψ_i satisfies both conditions, then the right endpoint of Ψ_i gets close to the intersection point of $f(\theta) = \theta + (|\Gamma_L| - L(\Psi_{i-1}))$ and $\Phi_U^*(\theta)$. Therefore, $\lim_{i \rightarrow \infty} |\Psi_i| = |\theta - L(\Gamma_L)|$ where θ is an angle such that $\Phi_U^*(\theta) = \theta + (|\Gamma_L| - L(\Psi_{i-1}))$. So the final orientation set can get arbitrarily close to this limit by increasing the number of push actions.

Recall that the symmetric case, where Γ_L is L-type, is similar. There exist imperfect parts [27] that meet both of the above conditions for some Θ_i .

6 Conclusion

In order for automated planning algorithms for part handling tasks to be useful in practice it is important that these algorithms are capable of dealing with the inevitable shape variations of real industrial parts. The few papers that do not assume perfect parts generally assume a very restrictive model for shape variations, and often only determine how big these variations must be to invalidate a solution that was computed

based on the perfect model part. In this paper, we have considered a more general model for shape variations and studied its effects on orienting parts by pushing. We have proposed an algorithm that takes into account these variations during planning and as such outputs a plan that simultaneously orients all instances satisfying the model into the smallest possible interval of orientations after a given number of push actions. We have also investigated the conditions for which the part cannot obtain the smallest final orientation set after finite number of push actions.

The set of possible orientations of an imperfect part can consists of several disjoint intervals. In this paper, we have focused on finding the smallest interval that contains all these subintervals. A different version of the problem would be to minimize the total size of the subintervals. Another extension is to allow for independent variations in the location of the center of mass. It is also interesting to explore which parameters affect the length of the final orientation interval; examples of such parameters are the width of the tolerance zone and the eccentricity of the part.

References

1. Donald, B.R.: Error Detection and Recovery in Robotics. Springer (1987)
2. Mason, M.T.: Manipulator Grasping and Pushing Operations. Ph.D. thesis, MIT, June 1982. Robot Hands and the Mechanics of Manipulation MIT Press (1985)
3. Chen, J., Goldberg, K., Overmars, M.H., Halperin, D., Böhringer, K.-F., Zhuang, Y.: Shape tolerance in feeding and fixturing. In: Agarwal, P.K., Kavraki, L.E., Mason M.T., Peters., A.K. (eds.) Robotics: the algorithmic perspective, pp. 297–311 (1998)
4. Lavalle, S.M.: Planning Algorithms, chapter 12: Planning Under Sensing Uncertainty. Cambridge University Press (2006)
5. Dogar, M., Srinivasa, S.S.: A framework for push-grasping in clutter. Robot.: Sci. Syst. 2 (2011)
6. Requicha, A.A.G.: Toward a theory of geometric tolerancing. Int. J. Robot. Res. 2, (1983)
7. Voelker, H.: A current perspective on tolerancing and metrology. Manufact. Rev. 6(4) 1993
8. Lozano-Perez, T., Mason, M.T., Taylor, R.H.: Automatic synthesis of fine-motion strategies for robots. Int. J. Robot. Res. 3(1), 3–24 (1984)
9. Erdmann, M.A., Mason, M.T.: An exploration of sensorless manipulation. IEEE J. Robot. Autom. 4, 367–379 (1988)
10. Akella, S., Mason, M.T.: Posing polygonal objects in the plane by pushing. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 2255–2262 (1992)
11. Goldberg, K.: Orienting polygonal parts without sensors. Algorithmica 10(2), 201–225 (1993)
12. Chen, Y.-B., Ierardi, D.J.: The complexity of oblivious plans for orienting and distinguishing polygonal parts. Algorithmica 14, 367–397 (1995)
13. Berretty, R.-P., Overmars, M.H., van der Stappen, A.F.: Orienting polyhedral parts by pushing. Comput. Geom.: Theor. Appl. 21, 21–38 (2002)
14. Wiegley, J.A., Goldberg, K., Peshkin, M., Brokowski, M.: A complete algorithm for designing passive fences to orient parts. Assembly Autom. 17(2), 129–136 (1997)
15. Berretty, R.-P., Goldberg, K., Overmars, M.H., van der Stappen, A.F.: Computing fence designs for orienting parts. Comput. Geom.: Theor. Appl. 10(4), 249–262 (1998)
16. Guibas, L., Salesin, D., Stolfi, J.: Epsilon geometry: building robust algorithms for imprecise computations. In: Proceedings of the 5th ACM Annual Symposium on Computational Geometry, pp. 208–217 (1989)
17. Davoodi, M., Mohades, A.: Data imprecision under λ -geometry: range searching problems. Scientia Iranica 20(3), 663–669 (2013)

18. Roy, U., Liu, C., Woo, T.: Review of dimensioning and tolerancing. *Compu. Aid. Desig.* **23**(7) (1991)
19. Panahi, F., van der Stappen, A.F.: Bounding the locus of the center of mass for a part with shape variation. In: *Proceedings of the Canadian Conference on Computational Geometry*, pp. 247–252 (2013)
20. Ostrovsky-Berman, Y., Joskowicz, L.: Tolerance envelopes of planar mechanical parts with parametric tolerances. *Comput. Aid. Desig.* pp. 531–534 (2005)
21. Bern, M., Eppstein, D., Guibas, L.J., Hershberger, J.E., Suri, S., Wolter, J.: The centroid of points with approximate weights. In: *Proceedings of the 3rd European Symposium Algorithms*, LNCS 979, pp. 460–472 (1995)
22. Akella, S., Mason, M.T.: Orienting toleranced polygonal parts. *Int. J. Robot. Res.* **19**(12), 1147–1170 (2000)
23. Brost, R.C.: Automatic grasp planning in the presence of uncertainty. *Int. J. Robot. Res.* **7**(1), 3–17 (1988)
24. Kehoe, B., Berenson, D., Goldberg, K.: Toward cloud-based grasping with uncertainty in shape: estimating lower bounds on achieving force closure with zero-slip push grasps. In: *Proceedings of the IEEE International Conference on Robotics and Automation* (2012)
25. Giora, Y., Kaplan, H.: Optimal dynamic vertical ray shooting in rectilinear planar subdivisions. In: *Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms*, pp. 19–28 (2007)
26. Rao, A.S., Goldberg, K.: Manipulating algebraic parts in the plane. *IEEE Trans. Robot. Autom.* **11**(4), 598–602 (1995)
27. Panahi, F., Davoodi, M., van der Stappen, A.F.: Orienting a part with shape variation Technical Report UUUCS-2014-007, Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands (2014)