ACM Symposium on Solid Modeling and Applications (2004)
G. Elber, N. Patrikalakis, P. Brunet (Editors)

Tolerance envelopes of planar mechanical parts

Y. Ostrovsky-Berman and L. Joskowicz

School of Engineering and Computer Science
The Hebrew University of Jerusalem, Jerusalem 91904, Israel
Email: yaronber@cs.huji.ac.il

Abstract

We present a framework for the systematic study of parametric variation in planar mechanical parts and for efficiently computing approximations of their tolerance envelopes. Part features are specified by explicit functions defining their position and shape as a function of parameters whose nominal values vary along tolerance intervals. Their tolerance envelopes model perfect form Least and Most Material Conditions (LMC/MMC). Tolerance envelopes are useful in many design tasks such as quantifying functional errors, identifying unexpected part collisions, and determining device assemblability. We derive geometric properties of the tolerance envelopes and describe four efficient algorithms for computing first-order linear approximations with increasing accuracy. Our experimental results on three realistic examples show that the implemented algorithms produce better results in terms of accuracy and running time than the commonly used Monte Carlo method.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computational Geometry and Object Modeling]: Curve, surface, solid, and object representations, J.6 [Computer-aided design (CAD)]: Computer Aided Tolerancing

1. Introduction

Manufacturing and assembly processes are inherently imprecise, producing parts that vary in size and form. The need to control the quality of the production and to manufacture parts interchangeably led to the development of toleranced specifications. Tolerance specifications are the critical link between the designer and the manufacturer. Designers prefer tight tolerances to ensure that the part will fit in the assembly and perform its function. Manufacturers, on the other hand, prefer loose tolerances to lower the production cost and decrease the need for quality machine tools, and precision measurement machines. Tolerance analysis methods play a key role in bridging between the two.

Tolerance allocation is difficult even to the most skilled of designers because it requires identifying the critical interactions of toleranced dimensions, which often have complex dependencies. Tolerancing methods have been developed and incorporated into most modern CAD software. Given a tolerance allocation, tolerance analysis consists of predicting the effect of the allowed variations on the design functions. Tolerance synthesis consists of finding tolerance intervals that meet the functional requirements at the lowest cost.

A key problem in tolerance analysis is computing the tolerance envelope of a part from its tolerance specification. Tolerance specifications define a family of parts consisting of all valid instances of the part. The tolerance zone of a part is the difference between the smallest volume containing all part instances and the largest volume contained in all part instances. Its boundaries, called the part tolerance envelopes, define the worst-case variability of the part features, and thus model perfect form Most and Least Material Conditions (MMC/LMC). Part tolerance zones are useful in design tasks such as quantifying functional errors, identifying unexpected part collisions, and determining device assemblability.

Recent research in Computer-Aided Tolerancing (CAT) describes methods for defining and computing tolerance zones for individual features from their tolerance specifications [Ame94, CRSV97, WGJ94]. However, many issues regarding tolerance zones for entire parts remain open: what is their geometric complexity, what are good approximations, and how to efficiently compute them. Previous works are limited by the descriptive power of their variational models, by the quality of the approximations they produce, and by their computational efficiency.

In this paper, we present a framework for the systematic study of parametric variation in planar mechanical parts and for efficiently computing approximations of their tolerance envelopes. The framework reflects current tolerancing practice, incorporates common tolerancing assumptions, and exposes the computational trade-offs. Of the two commonly used tolerance specification methods [Voe93], geometric and parametric, we chose parametric specification because it is best suited for functional tolerancing, can describe most geometric specifications and has a simple, mathematically well-defined semantics within which part variability can be studied analytically. In the proposed model, part features are specified by explicit functions defining their position and shape as a function of parameters whose nominal values vary along tolerance intervals. We derive geometric properties of the worst-case tolerance envelopes and describe four efficient algorithms for computing first-order linear approximations with successive accuracy. Our experimental results on three realistic examples show that the implemented algorithms produce simpler, faster, and/or more accurate results than the commonly used Monte Carlo method.

The remainder of this paper is organized as follows. In Section 2 we review previous methods for worst case tolerance analysis using tolerance zones and their counterparts in parametric spaces. Section 3 describes the parametric tolerancing model that we use, and Section 4 deals with the geometric properties of tolerance envelopes resulting from this model. Section 5 describes four algorithms for computing the tolerance envelopes with increasing accuracy, and Section 6 presents our experimental results. Section 7 concludes with a summary and a description of current and future work.

2. Previous work

Several models have been proposed to model parametric part variations. These include simple-shaped regions around boundary points [GCO*98, YC97] and fixed-distance boundary offsets [RR86, LWC97], which are computationally efficient but are often inaccurate since they ignore parameter dependencies. Pino et al [PBF01] describe a kinematic model to simulate the "motion" of the features tolerance zone but do not describe how to compute the entire part tolerance zone. Pasupathy and Wilhelm [PW01] model tolerance envelopes with B-splines and describe how to verify if a part is in tolerance without explicitly constructing its tolerance envelope. Bhide et al. [BDS01] use areal coordinates to describe the Tolerance Map, a convex volume of points corresponding to all possible locations and variations of a plane which can arise from geometrical tolerances on size, form, and orientation. Desrochers et al. [DBL03] use a similar concept, in which the location and orientation of the plane are described by screw parameters of small displacement. Both these methods enable stack up analysis in an assembly, but do not compute the corresponding volumes for the entire part. Sacks and Joskowicz [SJ98] developed a kinematic tolerance analysis method that computes contact tolerance zones of planar parametric parts in configuration space. The method uses a parametric part model similar to ours, and computes contact zones which are complementary to our part tolerance zones.

Several CAT packages provide tools for computing worst-case part tolerance zones [SvHK97]. Some compute tolerance zones from many randomly generated part shape instances drawn from a presupposed parameter distribution (the Monte Carlo method) [DS87, Gro76]. This is expensive and incomplete, as parts typically have hundreds of features defined by tens of parameters. ADAPT [Sch00], developed by Ford for internal use, computes the tolerance envelopes of parametric planar parts with procedural definitions [Hin94]. It has one procedure for each of the many feature definition cases and incorporates ad-hoc simplifying assumptions that preclude quantifying the approximation error. These drawbacks motivate our work.

$v_1 = (0, 0)$
$v_2 = (l_1, l_1 * \tan(\alpha_1))$
$v_3 = (l_1 + l_2, l_1 * \tan(\alpha_1) + l_2 * \tan(\alpha_2))$
$v_4 = (l_3 + l_4 * \cos(\alpha_5) - l_1 - l_2,$
$l_4 * \sin(\alpha_5) - l_1 * \tan(\alpha_4) - l_2 * \tan(\alpha_3))$
$v_5 = (l_3 + l_4 * \cos(\alpha_5) - l_1, l_4 * \sin(\alpha_5) - l_1 * \tan(\alpha_4))$
$v_6 = (l_3 + l_4 * \cos(\alpha_5), l_4 * \sin(\alpha_5))$
$v_7 = (l_3, 0)$

$l_1 = 30 \pm 0.3$mm	$r_1 = 40 \pm 0.2$mm
$l_2 = 20 \pm 0.3$mm	$r_2 = 35 \pm 0.2$mm
$l_3 = 100 \pm 0.5$mm	$r_3 = 30 \pm 0.3$mm
$l_4 = 120 \pm 0.5$mm	$r_4 = 30 \pm 0.3$mm
$\alpha_1 = 53.1° \pm 1°$	
$\alpha_2 = 26.5° \pm 1°$	
$\alpha_3 = 33.7° \pm 1°$	
$\alpha_4 = 18.4° \pm 1°$	
$\alpha_5 = 90° \pm 1°$	

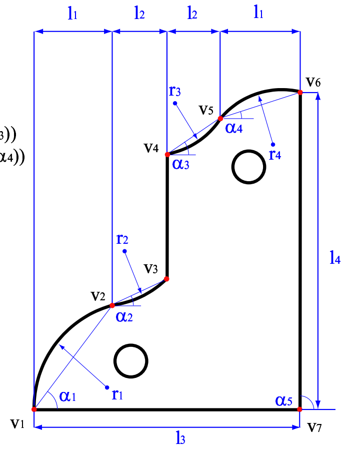


Figure 1: *Tolerance specification of a portion of a sewing machine cover. Vertices $v_1, \ldots, v_7$ are functions of subsets of the 13 parameters: lengths l_i, angles α_i and radii r_i. The part illustrated is the nominal part instance defined by the parameter vector $(\bar{l}_1, \ldots, \bar{l}_4, \bar{\alpha}_1, \ldots, \bar{\alpha}_5, \bar{r}_1, \ldots, \bar{r}_4)$.*

3. Tolerancing model

We propose the following tolerancing model for planar parts, which is very general in its semantics and has good computational properties.

Let A be a simple planar part whose boundary consists of curved segments. Its nominal shape and variation is defined by an m-dimensional parameter vector p. Each parameter has a nominal value and a tolerance interval, typically much smaller than the nominal dimension. Formally, the toleranced parametric part model is a 4-tuple $A = \langle V, S, \bar{p}, \Delta \rangle$, such that:

- $V = \{v_1(p), v_2(p), \ldots, v_n(p)\}$ is the **vertex set**, where each vertex $v_i(p) = (x_i(p), y_i(p))$, $1 \le i \le n$, is defined by an explicit standard elementary function of the **parameter vector** $p = (p_1, p_2, \ldots, p_m)$.
- $S = \{s_1, s_2, \ldots, s_n\}$ is the **segment set**, where the i^{th} segment is a curve connecting the vertices v_i, v_{i+1}, $1 \le i \le n-1$. Segment s_n connects the vertices v_n, v_1. A curve is a parametric function $s_i : [0,1] \times \Re^m \rightarrow E^2$, denoted by $s_i(\lambda, p)$. Certain curves depend on additional variables, such as Bézier control points, which are also functions of the parameter vector p.
- $\bar{p} = (\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_m)$ is the **nominal parameter vector**, where $\bar{p}_i$ is the nominal value of the i^{th} parameter, $1 \le i \le m$. The **offset vector** δ of parameter vector p is the offset from the nominal parameter vector $\bar{p}$, that is $\delta = (\delta_1, \delta_2, \ldots, \delta_m)$, where $\delta_i = p_i - \bar{p}_i$.
- $\Delta = \{(\delta_1^-, \delta_1^+), (\delta_2^-, \delta_2^+), \ldots, (\delta_m^-, \delta_m^+)\}$ is the **tolerancing set**, where δ_i^- and δ_i^+ are the minimal and maximal allowed variations from the nominal parameter value $\bar{p}_i$.

The **tolerance interval** of the i^{th} parameter is the interval: $\mathcal{P}_i = [\bar{p}_i + \delta_i^-, \bar{p}_i + \delta_i^+]$. The tolerance interval of the model is an m-dimensional hyperbox defined by the Cartesian product of the intervals: $\mathcal{P} = \mathcal{P}_1 \times \mathcal{P}_2 \times \ldots \times \mathcal{P}_m$. An **instance** $A(p)$ of the part model is the part defined by the parameter vector $p = (p_1, p_2, \ldots, p_m)$, where $p_i \in \mathcal{P}_i$, $1 \leq i \leq m$. Fig. 1 shows an example of a parametric part model of a portion of a sewing machine cover.

The tolerance envelopes of a point and of a segment of the part model are the boundaries of the union of all their instances. The outer and inner tolerance envelopes of a part model are the boundaries of the union and the intersection of all the instances, respectively.

The model can be directly generalized to parts with holes by treating each boundary separately. Note that the proposed model has the same semantics as the standard dimensional tolerancing scheme. Conventional tolerance drawings can be translated to the explicit functional representation. In the following, we assume that the parameters define geometrically valid part instances with the same topology and no self-intersections. When these assumptions do not hold, the specification describes physically unrealizable parts with no engineering meaning. Such invalid part models must be identified and reported to the engineer, so they can be fixed. The automatic validation of tolerance specifications is an important topic of current research [Ste93, APS98, KSD01].

4. Tolerance envelope properties

We now discuss the properties of the tolerance envelopes of individual segments. These form the basis for efficient computation of the part tolerance envelope.

We define the functions s_l, s_a, and s_b to parameterize line, arc, and Bézier curve segments, respectively. The parameter $\lambda \in [0,1]$ interpolates between the endpoints $v_1(p)$ and $v_2(p)$.

$$\begin{aligned} s_l(\lambda, p) &= (1-\lambda)v_1(p) + \lambda v_2(p) &(1)\\ s_a(\lambda, p) &= (1-\lambda)v_1(p) + \lambda v_2(p) + h(\lambda, p)v_{12}^{\perp}(p) &(2)\\ s_b(\lambda, p) &= \sum_{i=0}^{n-1} B_i^{n-1}(\lambda) b_i(p) &(3) \end{aligned}$$

where $h(\lambda, p) = \frac{-1+\sqrt{1+4\lambda\tan^2\frac{\alpha(p)}{2} - 4\lambda^2\tan^2\frac{\alpha(p)}{2}}}{2\tan\frac{\alpha(p)}{2}}$ is the height of the triangle connecting the segment $\overline{v_1(p)v_2(p)}$ to the arc point, $v_{12}^{\perp}$ is the normalized vector perpendicular to $v_2 - v_1$, $\alpha(p) \leq \pi$ is the arc angle, $b_i(p)$ are the Bézier control points such that $b_0(p) \equiv v_1(p), b_{n-1}(p) \equiv v_2(p)$, and $B_i^n(\lambda) = \binom{n}{i}\lambda^i(1-\lambda)^{n-i}$ are the Bernstein polynomials.

With today's manufacturing capabilities, tolerance intervals are usually at least two orders of magnitude smaller than nominal dimensions. Therefore, we use the standard first-order approximation and linearly approximate the vertex and segment functions around the nominal values. The linear approximation of vertex $v_i(p)$ is defined as:

$$v_i(p) \approx v_i(\bar{p}) + \sum_{j=1}^{m} \left(\frac{\partial v_i(\bar{p})}{\partial p_j}\right)\delta_j \qquad (4)$$

where $\delta_j \equiv (p_j - \bar{p}_j)$ is the j^{th} parameter offset and $\frac{\partial v_i(\bar{p})}{\partial p_j} \equiv \frac{\partial v_i(p)}{\partial p_j}|_{p=\bar{p}}$ is the partial derivative of $v_i(p)$ by parameter p_j evaluated at the nominal parameter value $\bar{p}$. Similarly, the linear approximations of line, arc, and Bézier curve segments are, respectively:

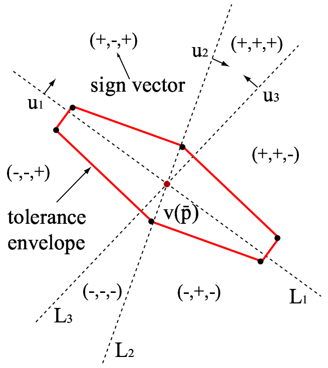


Figure 2: *Tolerance envelope of a point and its cone diagram with three non-zero partial parameter derivatives*

$$\begin{aligned} s_l(\lambda, p) &\approx s_l(\lambda, \bar{p}) + \sum_{j=1}^{m}\Big((1-\lambda)\frac{\partial v_1(\bar{p})}{\partial p_j} + \lambda\frac{\partial v_2(\bar{p})}{\partial p_j}\Big)\delta_j &(5)\\ s_a(\lambda, p) &\approx s_a(\lambda, \bar{p}) + \sum_{j=1}^{m}\Big((1-\lambda)\frac{\partial v_1(\bar{p})}{\partial p_j} + \lambda\frac{\partial v_2(\bar{p})}{\partial p_j} + \\ &\qquad h(\lambda, \bar{p})\frac{\partial v_{12}^{\perp}(\bar{p})}{\partial p_j} + \frac{\partial h(\lambda, \bar{p})}{\partial p_j}v_{12}^{\perp}(\bar{p})\Big)\delta_j &(6)\\ s_b(\lambda, p) &\approx s_b(\lambda, \bar{p}) + \sum_{j=1}^{m}\Big(\sum_{i=0}^{n-1} B_i^{n-1}(\lambda)\frac{\partial b_i(\bar{p})}{\partial p_j}\Big)\delta_j &(7) \end{aligned}$$

A key property of these approximations is that they depend only on the parameters that define the segment coordinates, which are those with non-zero partial derivatives. The number of such parameters, k_i, is usually much smaller than the total number of part model parameters m. In the following, k is the maximum number of dependent segment parameters.

Consider now the tolerance envelope of a point $v(\bar{p})$ on the part boundary. According to Equation 4, the displacement of $v(p)$ from the nominal point in a given direction d is $\langle v(p) - v(\bar{p}), d\rangle = \langle \sum_{j=1}^{m} u_j\delta_j, d\rangle$, where $u_j \equiv \frac{\partial v(\bar{p})}{\partial p_j}$. Thus the maximal displacement of a vertex v in direction d occurs at extremal parameter offset values δ_j^+ or δ_j^-. The sign of each parameter offset depends on the direction d and on the directions of the partial derivatives:

$$\delta_j^d = \begin{cases} \delta_j^- & \langle u_j, d\rangle < 0 \\ \delta_j^- & \langle u_j, d\rangle = 0 \text{ and } \langle u_j, d^{\perp}\rangle < 0 \\ \delta_j^+ & otherwise \end{cases} \qquad (8)$$

where $d^{\perp}$ is clockwise perpendicular to d and $\langle u_j, d\rangle$ is the vector inner product. Note that even when the derivative u_j is perpendicular to d, we assign an extremal value to δ_j. This ensures that $v(p)$ is a unique vertex of the tolerance envelope rather than a point on the boundary. Equation 8 shows that each non-zero partial derivative u_j divides the plane into two halves separated by a line L_j perpendicular to u_j, so that one half gets a maximal offset δ_j^+ and the other a minimal offset δ_j^-. We define L_j as passing through the nominal

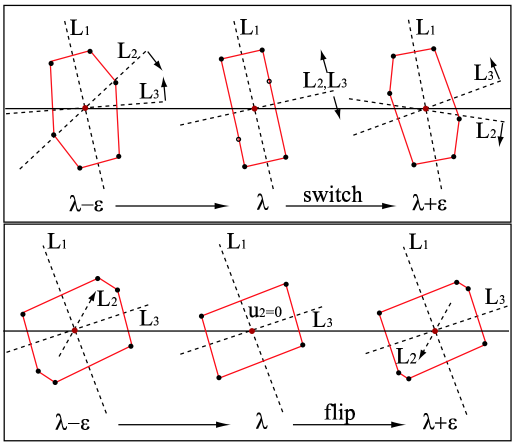


Figure 3: *Topological changes in the cone diagram. Cone diagram lines are dashed, and the tolerance envelope of the point is solid. The top figure shows a switch event, in which two cone diagram lines (L_2 and L_3) merge before switching position. The bottom figure shows a flip event, in which the line defined by u_2 changes orientation as a result of u_2 becoming zero.*

vertex $v(\bar{p})$, and oriented so that the positive offset sign is on its left.

The lines L_j induce a subdivision of the plane into $2k$ cones, which we call the **cone diagram** (Figure 2). The parameter offset signs for all directions d within a cone are the same, and define a vertex $v^d(p) = v(\bar{p}) + \sum_{j=1}^{m} u_j \delta_j^d$ which achieves the maximal displacement of $v(p)$ in the cone's directions. The following theorem summarizes the properties of the tolerance envelope of a point.

Theorem 1 Let A be a toleranced parametric part model and let $v(p)$ be a point on its boundary with k non-zero partial derivatives. Then the tolerance envelope of v is the boundary of a convex, centrally symmetric polygon with at most $2k$ vertices, and can be computed in optimal $O(k \log k)$ time.

Proof For each of the parameters p_i with non-zero derivative u_i, define the segment $\xi_i \equiv [u_i\delta_i^-, u_i\delta_i^+]$. Let $v(p_i) = v(\bar{p}_1, \bar{p}_2, \ldots, p_i, \ldots, \bar{p}_m)$, that is all parameters but p_i are at their nominal values. Observe that as p_i changes from $\bar{p}_i + \delta_i^-$ to $\bar{p}_i + \delta_i^+$ the point $v(p_i)$ moves along the segment ξ_i translated by $v(\bar{p})$, so it traces the set $\{v(\bar{p}) + x | x \in \xi_i\}$ which is the Minkowski sum of $v(\bar{p})$ and ξ_i. The Minkowski sum operation is commutative, so the tolerance envelope of v is the sum of $v(\bar{p})$ with all the segments ξ_i where $u_i \neq 0$. The Minkowski sum of segments is a convex centrally symmetric polytope called a zonotope [Zie94]. The complexity of a zonotope in the plane is at most twice the number of generating segments, that is $2k$. To compute the envelope, first sort the derivatives according to their angle with the x-axis, and construct the corresponding cone diagram. Then, choose an arbitrary cone and compute its sign vector and corresponding vertex according to Equation 8. Compute the next vertices by advancing the cones counter-clockwise, updating the vertex coordinates according to the parameter that inverts its sign. The running time is dominated by the sorting operation, which takes $O(k \log k)$. The algorithm is optimal because the angular sorting problem can be directly reduced to finding the tolerance envelope of a point. □

Conceptually, the tolerance envelopes of segments are the boundaries of the area swept by the point tolerance envelope as it moves along the nominal segment from one endpoint to another (Fig. 4). The tolerance envelope of a point on a nominal segment is computed as for vertices, except that the vectors u_j are now functions of λ (the terms in brackets in Eqs. 5, 6, and 7). As the point moves between the segment endpoints, its cone diagram lines rotate, thus changing the shape of the tolerance envelope. When the cone diagram lines overlap or change their orientation, the topology of the diagram (and therefore of the point envelope) changes (Fig. 3). This observation is true for all segment types. The only difference is the equations that determine the topological events, and the type of curves traced by the point envelop vertices as they sweep from one endpoint to the other. We detail the properties of the resulting envelope in Section 5.4.

5. Tolerance envelope computation and approximation

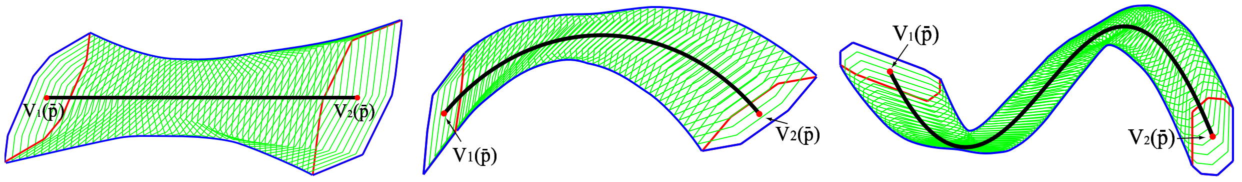
We now address the representation and computation of part tolerance envelopes. We are interested in part tolerance envelopes whose segments are as few and as simple as possible, yet are as close as possible to the real boundary. Simple segments follow the perfect form assumption, which stipulates that the tolerance envelope of a segment is a chain of segments of the same type, and facilitates further analysis and manipulation, such as stack-up tolerance analysis, collision detection, and assembly planning. Fewer segments speed up computation but compromise accuracy and simplicity, e.g., one long spline curve versus many short line segments. Since different applications will require different trade-offs between shape simplicity, accuracy, and efficiency, we define four successive approximations for segment tolerance envelopes: 1. Vertex Envelope Approximation (VEA); 2. Extremal Parameter Approximation (EPA); 3. Extremal Vertex Approximation (EVA); 4. Best Segment Approximation (BSA). We describe them next.

5.1. Vertex Envelope Approximation (VEA)

This is the simplest approximation, connecting the tolerance envelopes of neighboring vertices with two segments. The tolerance envelope of a line segment is the boundary of the convex hull of the points on the tolerance envelopes of its two endpoints. It is computed in linear time by finding the outer tangents of the endpoint envelopes. The convex hull contains all the instances of the line segment, and therefore contains its tolerance envelope. We say that this envelope is conservative, because it never underestimates the worst case behavior of the part. There is no analogue for the convex hull in circular and Bézier segments, so for these one of the other approximations must be used.

5.2. Extremal Parameter Approximation (EPA)

The vertices of the tolerance envelope of the starting endpoint have unique offset vectors that define instances of the toleranced segment. These instances define the starting paths of the vertices in the sweeping of the tolerance envelope described at the end of Section 4. Similarly the vertices of the tolerance envelope of the other endpoint define the end of the sweep paths. The instances correspond to the starting and ending topology of the tolerance envelope, and if there are no additional topological events in between, then the upper and lower envelopes of the arrangement they define is a good



(a) line segment (b) circular arc (c) Bézier curve

Figure 4: *Tolerance envelopes of individual segments. Solid thick lines represent the nominal segments, solid thin lines tolerance envelopes, and dotted polygons instances of point tolerance envelopes along the segment.*

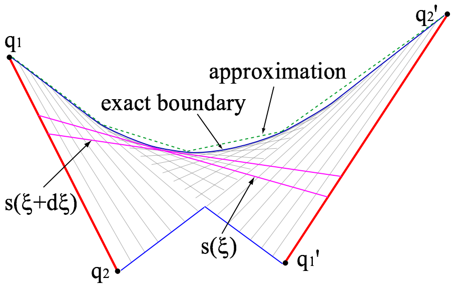
approximation to the upper and lower parts of the tolerance envelope. We find the upper and lower envelopes of the arrangement with a divide and conquer technique [Her89].

5.3. Extremal Vertex Approximation (EVA)

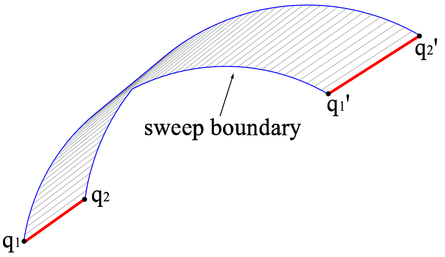
The tolerance envelope defined by the interpolation parameter λ is the boundary of a convex polygon containing all the points $s(\lambda, p)$ with $p \in \mathcal{P}$. Except for specific values of λ (see below), there is a unique polygon vertex that attains the maximal displacement from the nominal point $s(\lambda, \bar{p})$ in the direction d orthogonal to the tangent to the nominal segment at $s(\lambda, \bar{p})$. We call this vertex an **extremal vertex**. This vertex has the offset vector of the cone that contains the direction d, and as we change λ it will trace a path on the instance defined by this offset vector. The offset vector of the extremal vertex changes only when the cone containing d changes. We call this an **extremal event**. The values of λ at which extremal events occur are solutions to the equation $\langle u_i(\lambda), d^{\perp}(\lambda)\rangle = 0$ (there are $O(k)$ such events). In extremal events λ_e, the tolerance envelope polygon has an edge parallel to the tangent at $s(\lambda_e, \bar{p})$. One endpoint of this edge is the extremal vertex of $\lambda < \lambda_e$, and the other is the next extremal vertex. Note that because of the symmetry of the point tolerance envelope, there is a symmetrical vertex for each extremal vertex (from the other side of the nominal segment). The EVA extends EPA by computing the extremal events and the instances of segments corresponding to all the extremal vertices along the sweep. It computes the upper and lower envelope of the arrangement defined by these instances.

5.4. Best Segment Approximation (BSA)

This method computes the tolerance envelope of a segment up to the desired degree of accuracy. For applications that support general algebraic curves, it computes the exact tolerance envelope (after the linear approximation of the segment). The algorithm performs the sweep of the tolerance envelope from one endpoint to the other. A key issue is the identification of discrete events in which the topology of the cone diagram changes (Fig. 3). The values of λ in which topological changes occur are solutions to the equations $\langle u_i(\lambda), u_j^{\perp}(\lambda)\rangle = 0$ and $u_i(\lambda) = 0$, where $u_j^{\perp}(\lambda)$ is perpendicular to $u_j(\lambda)$. The first equation corresponds to two cone diagram lines coinciding. We call this a **switch event**. The second equation corresponds to a cone diagram line disappearing before changing its orientation. We call this a **flip event**. Flip events are degenerate cases that occur frequently in practice. They are equivalent to k simultaneous switch events, but we treat them separately for efficiency and robustness. There are $O(k^2)$ switch events and $O(k)$ flip events. Between topological events, each vertex of the point envelope has a constant parameter offset vector, and therefore moves on a curve of the same type as the nominal segment.



(a) line sweeps



(b) circular arc sweeps

Figure 5: *Boundary of segment sweeps between two consecutive topological events. (a) the edge endpoints move along linear paths. The figure shows both the exact boundary of the swept area (solid) and the approximation with four line segments (dashed); (b) the endpoints move along circular arcs*

A basic step in the algorithm is to compute the boundary of the area swept by two neighboring point envelope vertices (an edge) from one topological event to the next. Figure 5 shows the area swept by an edge moving along line and arc segment paths. The boundary of the swept area consists of a chain of segments of the nominal type and a general curve. The general curve is obtained in one of the following methods:

1. Using the general technique of Kim et al. [KA93] that computes the boundary of a general time dependant curve swept along a general trajectory. The complexity of this method depends on the degree of the segment curve and the size of the output. In our case these are small constants.
2. Using a closed form expression for the special case of line and arc sweeps. The expression is derived as follows. Let $s(p)$ de-

```
1.  Solve the topological event equations $\langle u_i(\lambda), u_j^{\perp}(\lambda)\rangle = 0$ and $u_i(\lambda) = 0$
2.  Sort the solutions in increasing order
3.  Compute the offset vectors of the starting tolerance envelope and store their corresponding segment instances
4.  For each topological event:
    - If switch event $(\lambda, i, j)$
        a. Sweep the four edges defined by $(i, j)$ from previous to current $\lambda$
        b. Update the two middle cones' instances by flipping the $i, j$ parameter offset signs
    - If flip event $(\lambda, i)$
        a. Sweep all the edges from previous to current $\lambda$
        b. Update all the instances by flipping the $i^{th}$ parameter offset sign.
5.  Compute the upper and lower envelopes of the segments from all the sweeps
```

Table 1: *The Best Segment Approximation (BSA) algorithm of a single segment.*

note the toleranced segment instance with parameter vector p, and let p_{q1} and p_{q2} denote the parameter vectors of the endpoints of the swept edge (note that they differ in one parameter sign only). Define the parameter vector $p(\xi) = (1-\xi)p_{q1} + \xi p_{q2}$. The general curve in the swept boundary is determined by the intersection of instances with infinitesimal difference in the parameter vector (see Figure 5a). Compute the expression for the intersection of $s(p(\xi))$ with $s(p(\xi + d\xi))$ (there are two solutions for arc segments), and take the limit as $d\xi \rightarrow 0$. The resulting expression is the desired parametric curve with parameter ξ. The curve is clipped at the values of ξ corresponding to the intersections with the extremal instances $s(p_{q1})$ and $s(p_{q2})$. We followed the above derivation steps using Maple and derived a polynomial curve of second degree for the linear sweep, and a non-rational curve of second degree for the circular sweep.

3. Approximating the curve without explicitly computing it by uniformly sampling a constant number of instances $s(p(\xi))$ with values of ξ in $[0,1]$, and computing their upper and lower boundary.

For applications that require perfect form tolerance envelopes, we approximate the swept curves either directly as in method 3 above, or by sampling a constant number of points on the curve using a variation on the Douglas-Peucker heuristic [DP73] for continuous curves. In our implementation, we approximated the curve with five segments using method 3.

Table 1 summarizes the BSA algorithm. The algorithm starts by calculating the topological events and sorting them by λ values. It then iterates over the events, computing the upper and lower envelopes of the area swept by the edges that participate in the current topological change. Switch events require $O(1)$ for update, as they affect only six cones (three symmetrical ones). Flip events affect all the cones and require $O(k)$ time for sweeping all the edges. These steps produces $O(k^2)$ segments and curves, although not all contribute to the tolerance envelope. The last step computes the upper and lower envelopes of these segments in $O(k^2 \log k)$. The complexity of the lower and upper envelopes, which dominates the combinatorial complexity of the actual segment tolerance envelope (under the linear approximation) is $O(k^2)$.

Approximation	Space	Time	Conservative
VEA	$O(1)$	$O(k \log k)$	yes
EPA	$O(k)$	$O(k \log k)$	no
EVA	$O(k)$	$O(k^2)$	no
BSA	$O(k^2)$	$O(k^2 \log k)$	yes

Table 2: *Properties of the approximation algorithms per segment. The actual space and time upper bounds include a multiplicative factor due to the complexity of Davenport-Schinzel sequences [SA95], which, for all practical cases, can be treated as a small constant. For the time complexity, we assume that the intersection of Bézier curves of low degree is found in $O(1)$ time. The overall complexity bound is n times the complexity of the approximation algorithm.*

Table 2 summarizes the computational properties of the approximation algorithms. The VEA is the least accurate but gives good results when there are no extremal events in the cone diagram. EPA gives tighter results but misses segment instances that may contribute to the envelope when more than one topological event occurs. The EVA improves on EPA at the cost of time complexity by tracking extremal changes without calculating topological events. BSA is the most computationally expensive but gives both accurate and conservative results. Let $approx(A)$ denote the area bounded by the outer tolerance envelope produce by the corresponding approximation of part A, then the relation between the four approximations is $EPA(A) \subseteq EVA(A) \subseteq BSA(A) \subseteq VEA(A)$. The relation is reversed for the inner tolerance envelope.

5.5. Tolerance envelope of the part

The tolerance envelope of the entire part is computed by merging the tolerance envelopes of its segments. Consecutive segments have one common vertex, at the envelope of which their segment envelopes terminate. When the segment envelope chains do not intersect, we merge them with vertices on the common vertex envelope. When they do intersect, we find the intersection with a segment intersection algorithm [Bal95] in $O(c \log c)$ time, where c is the length of both chains, and merge the chains at the intersection point. If there is more than one intersection then the envelope has a hole or self intersects. In this case the algorithm informs the user that the part model is invalid and must be fixed.

6. Experimental results

We implemented the four algorithms for parts composed of line and arc segments. The implementation was written in C++ with the CGAL library, and run on a 2.4 GHz Pentium 4 with 1GB RAM

model	BSA					EPA					Monte Carlo				
	relative(%)		value(mm)		time	relative(%)		value(mm)		time	relative(%)		value(mm)		time
	$\bar{x}$	σ	$\bar{x}$	σ	(ms)	$\bar{x}$	σ	$\bar{x}$	σ	(ms)	$\bar{x}$	σ	$\bar{x}$	σ	(ms)
cover	1.71	1.30	0.023	0.019	406	1.83	2.08	0.023	0.020	172	16.7	10.9	0.294	0.274	1938
handbrake	0.23	0.32	0.001	0.001	687	0.23	0.32	0.001	0.001	437	12.7	10.2	0.071	0.080	5672
support	0.52	0.47	0.007	0.005	594	5.91	14.6	0.162	0.478	391	22.7	15.9	0.571	0.531	3100
cover / 2	0.81	0.60	0.005	0.004	411	0.85	0.97	0.005	0.005	179	18.3	11.2	0.157	0.135	1951

Table 3: *Approximation error statistics for the BSA, EPA, and Monte Carlo envelopes (averaged over 10 random sampling of 100 instances each), on four examples. For each entry, the first two columns show the deviation of the relative error (mean and standard), the following two the deviation of the absolute value in millimeters, and the last one the running time in milliseconds.*

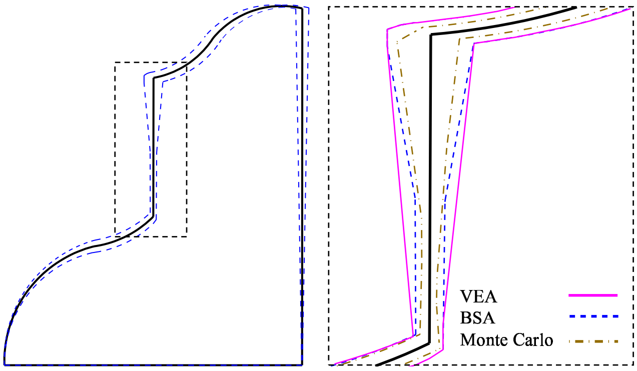
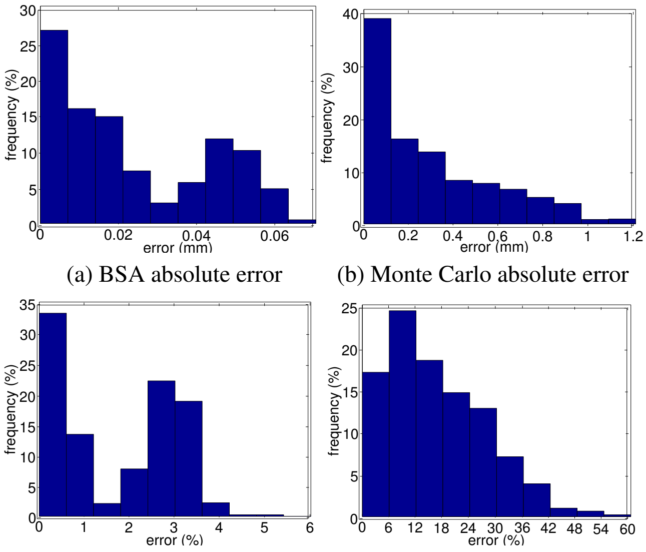


Figure 6: *The sewing machine cover tolerance envelope and an enlarged detail of it. Solid thick curves are the nominal part boundary. Thin solid, dashed, and alternating dashed curves are VEA, BSA, and Monte Carlo envelopes, respectively.*

under Windows XP. To empirically quantify the accuracy of the approximations, we compared them with a dense sampling of the envelope computed without the linear approximation. For each sampled point, we compute its parametric function and solve the nonlinear optimization problem consisting of maximizing the offset in the normal direction. In the following examples, we uniformly sampled each segment with 200 points. Running times were 15-45 minutes on MATLAB (the MATLAB profiler shows that 54.5% of the running time is spent inside optimized built in C code, so the running time can be cut in half at the most, still much too slow for interactive tolerance analysis).

We compare the results of our algorithms with those produced by the Monte Carlo sampling method used in most CAT systems (CATIA, TASys, TolStack). We computed part envelopes by randomly generating part instances and then computing the outer envelopes that minimally contain the instances, and the inner envelopes that are maximally contained in them. We call the resulting envelopes Monte Carlo envelopes. The part instances are generated by randomly choosing parameter values from a uniform distribution, computing the upper and lower envelope of the instances of each segment, and merging the chains. In the following experiments, we generated 100 part instances per Monte Carlo envelope.

Fig. 6 shows the results of the algorithms on the the sewing machine cover. Note that the VEA envelope is too conservative, while the Monte Carlo envelope is too optimistic, missing the instances that cause extremal offsets from the nominal. Fig. 7 shows a detailed comparison between the BSA and the Monte Carlo envelopes. Note that BSA envelope is much more accurate, with over 95% of the envelope under 0.057mm and 3.52% relative error.



(a) BSA absolute error (b) Monte Carlo absolute error

(c) BSA relative error (d) Monte Carlo relative error

Figure 7: *Error distribution of the envelope approximations of the sewing machine cover. The horizontal axis is the difference between the exact and the approximated distance from the nominal boundary. The vertical axis is the percentage of the envelope error. The top two graphs show the absolute difference value in millimeters, while the bottom two show the relative difference value from the actual distance.*

Table 3 shows approximation error statistics of the BSA, EPA, and Monte Carlo envelopes on the sewing machine cover, a part of a handbrake mechanism, and an axis support models (Fig. 8). In all cases, the BSA has a mean and standard deviation error value with the same order of magnitude as the tolerance interval squared, as expected from a linear approximation. On average, the Monte Carlo envelope of a hundred instances is ten times less accurate and runs ten times slower than BSA and EPA. We also ran the algorithms on the sewing machine cover model with tolerance intervals tightened by a factor of 2 (cover / 2). The effect on the error absolute value of all the approximations is the expected scaling down. Notice, however, that for BSA and EPA, the relative error is also significantly lower, with BSA maximal error of 3.01% (0.0147mm). This demonstrates that the linear approximation becomes better as

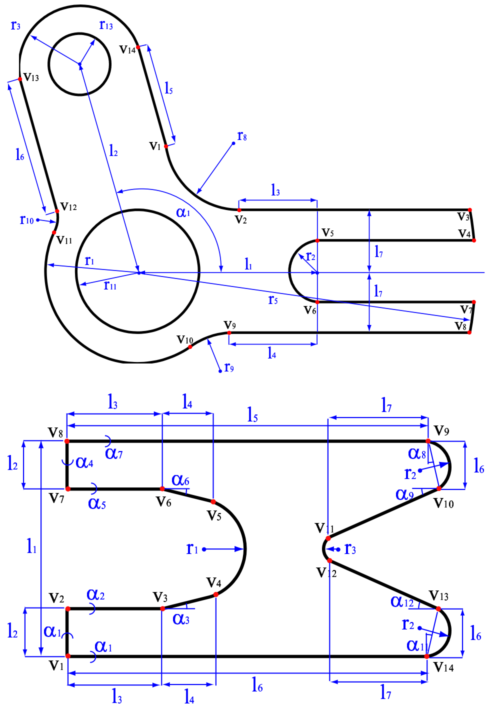


Figure 8: *Tolerance specification of a handbrake part (up) and an axis support part (down) modelled with 16 vertices and 22 parameters, and 14 vertices and 19 parameters, respectively.*

the tolerances become tighter. On the other hand, the Monte Carlo envelope has the same relative error.

7. Conclusion

We have presented a framework for modeling parametric variation in planar parts with curved boundaries and for efficiently computing first-order approximations of their worst-case tolerance envelopes. Based on the geometric properties of the tolerance envelopes that we derived, we developed four efficient algorithms that trade-off between shape simplicity, accuracy, and efficiency. Their complexity ranges from $O(n)$ space and $O(nk \log k)$ time complexity for the Vertex Envelope Approximation, to $O(nk^2)$ space and $O(nk^2 \log k)$ time complexity for the Best Segment Approximation, where n is the number of boundary segments and k is the maximum number of dependent segment parameters, which is usually much smaller than the total number of part parameters. The algorithms offer clear running time, simplicity, and accuracy advantages over the commonly used Monte Carlo method, as demonstrated by our experimental results on three realistic examples.

We are currently investigating the use of tolerance envelopes in a variety of mechanical design and assembly planning tasks, including tolerance envelope stack-up in chains of mated parts, assembly planning with tolerance parts, and toleranced configuration space computation [SJ98]. We are also planning to develop algorithms for statistical part tolerance envelope computation, and for spatial parts.

Acknowledgement: This work was partially supported by grant from the Israeli Ministry of Science, Grant number 01-01509.

References

[Ame94] The American Society of Mechanical Engineers: *ASME Y14.5M-1994 Mathematical Definition of Dimensioning and Tolerancing Principles*. New York, 1994. 1

[APS98] Andersson L.-E., Peters T. J., Stewart N. F.: Selfintersection of composite curves and surfaces. *Computer Aided Geometric Design 15*, 5 (1998), 507–527. 3

[Bal95] Balaban I. J.: An optimal algorithm for finding segments intersections. In *Proc. of the 11th annual symposium on Computational Geometry* (1995), pp. 211–219. 6

[BDS01] Bhide S., Davidson J. K., Shah J. J.: Areal coordinates: The basis of a mathematical model for geometric tolerances. In *Proc. of the 7th CIRP Int. Seminar on Computer-Aided Tolerancing* (Paris, France, 2001), pp. 35–44. 2

[CRSV97] Clément A., Rivière A., Serré P., Valade C.: The TTRS: 13 constraints for Dimensioning and Tolerancing. In *Proc. of the 5th CIRP Int. Seminar on Computer-Aided Tolerancing* (Toronto, 1997). 1

[DBL03] Desrochers A., Béron V., Laperrière L.: Revisiting screw parameter formulation for accurate modeling of planar tolerance zones. In *Proc. of the 8th CIRP Int. Seminar on Computer-Aided Tolerancing* (Charlotte, NC, 2003), pp. 239–248. 2

[DP73] Douglas D. H., Peucker T. K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Canadian Cartographer 10*, 2 (1973), 112–122. 6

[DS87] DeDoncker D., Spencer A.: Assembly tolerance analysis with simulation and optimization techniques. *SAE Transactions 96*, 1 (1987), 1062–1067. 2

[GCO*98] Goldberg K., Chen J., Overmars M., Halperin D., Karl B., Y. Z.: Shape tolerance in feeding and fixturing. In *Robotics, The Algorithmic Perspective*, Agarwal P., Kavraki L.,, Mason M., (Eds.). A.K Peters, The Netherlands, 1998. 2

[Gro76] Grossman D.: *Monte Carlo simulation of tolerancing in discrete parts manufacturing and assembly.* Tech. Rep. STAN0CS-76-555, Stanford AI laboratory, 1976. 2

[Her89] Hershberger J.: Finding the upper envelope of n line segments in $O(n \log n)$ time. *Information Processing Letters 33* (1989), 169–174. 5

[Hin94] Hinze C. U.: *A Contribution to Optimal Tolerancing in 2-Dimensional Computer Aided Design.* PhD thesis, Johannes Kepler Universität Linz, 1994. 2

[KA93] Kim M.-S., Ahn J.-W.: An algebraic algorithm to compute the exact general sweep boundary of a 2D curved object. *Information Processing Letters 47*, 5 (1993), 221–229. 5

[KSD01] KANDIKJAN T., SHAH J. J., DAVIDSON J. K.: A mechanism for validating Dimensioning and Tolerancing schemes in CAD systems. *Computer Aided Design 33* (2001), 721–737. 3

[LWC97] LATOMBE J.-C., WILSON R., CAZALS F.: Assembly sequencing with toleranced parts. *Computer-aided Design 29*, 2 (1997), 159–174. 2

[PBF01] PINO L., BENNIS F., FORTIN C.: Calculation of virtual and resultant parts for variational assembly analysis. In *Proc. of the 7th CIRP Int. Seminar on Computer-Aided Tolerancing* (Paris, France, 2001), pp. 83–92. 2

[PW01] PASUPATHY T. M. K., WILHELM R.: Curves for profile tolerance zone boundaries. In *Proc. of the 7th CIRP Int. Seminar on Computer-Aided Tolerancing* (Paris, France, 2001), pp. 45–54. 2

[RR86] ROSSIGNAC J. R., REQUICHA A. A. G.: Offseting operations in solid modelling. *Computer Aided Geometric Design 3* (1986), 129–148. 2

[SA95] SHARIR M., AGARWAL P.: *Davenport-Schinzel Sequences and Their Geometric Applications*. New York: Cambridge University Press, 1995. 6

[Sch00] SCHULTHEISS R., FORD WERKE AG: *ADAPT Users Manual: Computer-Aided Tolerance Analysis and Design*. Koln, Germany, 2000. 2

[SJ98] SACKS E., JOSKOWICZ L.: Parametric kinematic tolerance analysis of general planar systems. *Computer-Aided Design 30*, 9 (1998), 707–714. 2, 8

[Ste93] STEWART N. F.: Sufficient condition for correct topological form in tolerance specification. *Computer Aided Design 25*, 1 (1993). 3

[SvHK97] SOLOMONS O. W., VAN HOUTEN F., KALS H.: Current status of CAT systems. In *Proc. of the 5th CIRP Int. Seminar on Computer-Aided Tolerancing* (Toronto, 1997). 2

[Voe93] VOELCKER H.: A current perspective on Tolerancing and Metrology. *Manufacturing Review 6*, 4 (1993), 258–268. 1

[WGJ94] WHITNEY D., GILBERT O., JASTRZEBSKI M.: Representation of geometric variations using matrix transforms for statistical tolerance analysis. *Research in Engineering Design 6*, 4 (1994), 191–210. 1

[YC97] YAP C. K., CHANG E.-C.: Issues in the Metrology of Geometric Tolerancing. In *Algorithms for Robot Motion Planning and Manipulation* (Wellesley, Massachusetts, 1997), Laumond J.-P., Overmars M., (Eds.), A.K. Peters, pp. 393–400. 2

[Zie94] ZIEGLER G. M.: *Lectures on Polytopes*, vol. 152 of *Graduate Texts in Mathematics*. Springer-Verlag, Heidelberg, 1994. 4