Graphical Models and Image Processing **61,** 341–361 (1999)
Article ID gmip.1999.0507, available online at http://www.idealibrary.com on IDEAL®

On Computing Contact Configurations of a Curved Chain

Kai Tang

Unigraphics Solutions Inc., 4251 Plymouth Road, P.O. Box 986, Ann Arbor, Michigan 48106-0986
E-mail: tangk@ugsolutions.com

Received December 7, 1998; revised August 26, 1999; accepted August 26, 1999

Given a simple generalized polygon $\mathcal{A}$ of line segments and arcs that is free to move and rotate and an oriented monotone chain $\mathcal{B}$ composed of smooth parametric curved edges, the positions and orientations for $\mathcal{A}$ to gouge-freely contact $\mathcal{B}$ (i.e., the contact configurations) is a C^0 continuous surface in a three dimensional space $\mathbf{R}^3$. Past results either limit $\mathcal{B}$ to be polygonal or depend on the very complicated cylindrical algebraic decomposition algorithm, which is difficult to implement in practice and does not apply to parametric curves. We address this problem by conducting a rigorous study of the geometric and topological structures of the contact configurations surface and providing the exact mathematical descriptions of the faces, edges, and vertices on this surface. A practical intersection algorithm is proposed for computing the critical curves on the contact configurations surface. In addition, an application of the contact configurations in mill-turn machining is presented.

Key Words: gouge-free contact; configuration space; ruled surface; offsets; contact maps.

1. INTRODUCTION

This paper studies the problem of characterizing the gouge-free contact between a free simple generalized polygon and an oriented curved chain in the plane, and its applications. Formally, let $\mathcal{A}$ be a compact (i.e., bounded and closed) subset in the X–Y plane whose boundary is a simple Jordan curve made of line segments and arcs; let g and l be a designated reference point and a reference vector in $\mathcal{A}$, respectively. An instance of $\mathcal{A}$, to be denoted as $\mathcal{A}(x, y, \theta)$, is completely described by the triplet (x, y, θ), where (x, y) is the X–Y coordinates of g and θ is the counterclockwise angle measured from the $+X$ axis to l. Let $\mathcal{B}$ be a connected chain in the plane that is composed of smooth curves and is monotone with respect to the X-axis. The *interior* of $\mathcal{B}$ is defined as the set $\mathcal{B}_< = \{(x, y) \mid \exists\,(x, y') \in \mathcal{B} \rightarrow y' > y\}$. The following definition is in order.

1077-3169/99 $30.00

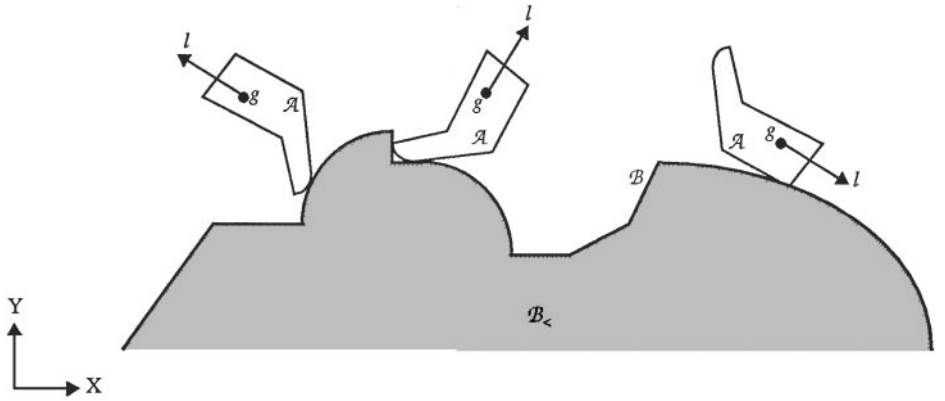


FIG. 1.1. Contact configurations of $\mathcal{A}$ and $\mathcal{B}$.

DEFINITION. A triplet (x, y, θ) is called a contact configuration of $\mathcal{A}$ and $\mathcal{B}$, if and only if $\mathcal{A}(x, y, \theta) \cap \mathcal{B} \neq \varnothing$ and $\mathcal{A}(x, y, \theta) \cap \mathcal{B}_< = \varnothing$. The set of all contact configurations of $\mathcal{A}$ and $\mathcal{B}$ is referred to as the contact configuration space of $\mathcal{A}$ and $\mathcal{B}$, denoted as $\mathcal{C}_{\mathcal{A},\mathcal{B}}$. Refer to Fig. 1.1 for an example of these definitions.

The contact configuration space belongs to a broader and well-studied problem in robot motion planning—the construction of the free space of a robot amid a set of obstacles in two- and three-dimensional spaces. In our case, which is two-dimensional, $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ is exactly the boundary of the free space of $\mathcal{A}$ with $\mathcal{B}$ being the obstacle. Abundant research results notwithstanding (for example, see [1, 2, 4, 6, 8–10, 12–15, 18] and Latombe's excellent book [7]), we believe that a careful investigation of $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ in our setting is still well needed, based on the following three reasons.

(1) Most published reports dealing with $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ assume that $\mathcal{B}$ is polygonal, i.e., represented by line segments only. To use their algorithms to achieve accurate results, a curved $\mathcal{B}$ needs to be approximated by a large number of tiny linear segments, which usually cause the $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ to have very long and skinning faces. Our experiments show that building such a $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ is extremely sensitive to floating precisions and geometric tolerances, in addition to large amount of computing time brought by the linear approximation.

(2) Even if the cylindrical decomposition method [7, 14] provides a theoretical tool for characterizing $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ for a $\mathcal{B}$ with algebraic curves, it is usually only used to find a free path without actually and explicitly constructing $\mathcal{C}_{\mathcal{A},\mathcal{B}}$. In some applications, such as the one given in this paper, a complete and explicit $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ is required. It is not clear how the decomposition algorithm can help in this regard.

(3) Finally, the curves on $\mathcal{B}$ are not always algebraic. Quite the opposite, they are usually given as parametric curves, such as in CAD/CAM. The contact analysis of analytic shapes (cf. [4])—sphere, plane, cylinder, line, circle—is not applicable.

In this paper, we present a rigorous study of the exact mathematical descriptions and properties of the entities of $\mathcal{C}_{\mathcal{A},\mathcal{B}}$, i.e., its faces, edges, and vertices, based directly on the original mathematical definition of $\mathcal{B}$ without linear approximation. The only limitation we require on $\mathcal{A}$ and $\mathcal{B}$ is that they can only be locally nonconvex.[1] Geometrically, this

[1] A compact subset $\mathcal{S}$ in the plane is locally nonconvex if there exist only a finite number of points p such that the intersection of $\mathcal{S}$ with a convex neighborhood of p (e.g., a disc) is nonconvex [7].

means that the arcs and curves on $\mathcal{A}$ and $\mathcal{B}$ must be convex, e.g., the example in Fig. 1.1. (A curved edge E on $\mathcal{B}$ is convex if and only if the curvature vector κ of every point on E points toward the interior of $\mathcal{B}$. That is, the inner product between κ and the $-Y$-axis is nonnegative.)

The paper is organized as follows. First, we present the result of the fundamental case—when both $\mathcal{A}$ and $\mathcal{B}$ are convex. Detailed mathematical descriptions are given for the faces, edges, and vertices on the contact configuration space. Next, we relax $\mathcal{B}$ to be monotone (with respect to the X-axis) and propose a construction algorithm of $\mathcal{C}_{\mathcal{A},\mathcal{B}}$, based on the trimming-and-enveloping concept. An intersection algorithm is described for computing the critical curves on $\mathcal{C}_{\mathcal{A},\mathcal{B}}$. We then give a general account on how $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ can be constructed when $\mathcal{A}$ is relaxed to be nonconvex. Finally, we give an application example of $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ in numerically controlled machining: how to find a minimum number of orientations of the tool that accomplish the machining task, followed by the conclusion remarks.

Before preceding to the next section, the following notations are defined.

$\mathcal{A}$. A simple generalized polygon represented as a counterclockwise ordered list of points (vertices) $\{a_1, a_2, \ldots, a_m\}$, with edges $\{e_i = \langle a_i, a_{i+1(\mathrm{mod}\, m)}\rangle : i = 1, 2, \ldots, m\}$. Each e_i is either a line segment or a convex circular arc. The *differential angle* of a_i, denoted as Δ_i, is described as the difference in the slope angles between the tangent vectors of e_i and e_{i-1} at a_i, as shown in Fig. 1.2a.

$\mathcal{B}$. A simple connected chain of curves that is monotone with respect to the X-axis. $\mathcal{B}$ is represented by its list of points (vertices) $\{b_1, b_2, \ldots, b_n\}$ and edges $\{E_i = \langle b_i, b_{i+1}\rangle : i = 1, 2, \ldots, n-1\}$. Each E_i is a C^1 continuous and convex parametric curve $E_i(t) = (x_i(t), y_i(t)) : t \in [0, 1]$, with its start and end points on b_i and b_{i+1}, respectively. Each vertex b_i, except for b_1 and b_n, is associated with two tangent vectors, the *left tangent* $\frac{d}{dt}E_{i-1}(t)|_{t=1}$ and the *right tangent* $\frac{d}{dt}E_i(t)|_{t=0}$, whereas at the two boundary vertices, the left tangent of b_1 is the $+Y$ axis while the right tangent of b_n is the $-Y$ axis.

$\omega_i(t)$: The slope angle function of edge $E_i(t)$. This is the angle measured from the $+X$ axis to the tangent of E_i at parameter t. (See Fig. 1.2b.) It is both C^1 continuous and monotone due to the smoothness and convexity of $E_i(t)$.

$\mathbf{R}_\alpha$: The rotation operator in the plane. Given a point (x, y), the application of $\mathbf{R}_\alpha$ on it, i.e., $(x, y) \cdot \mathbf{R}_\alpha$, yields a new point $(x', y') = (x \cdot \cos\alpha - y \cdot \sin\alpha, x \cdot \sin\alpha + y \cdot \cos\alpha)$. That is, (x, y) is rotated about the origin counterclockwise by an angle α. See Fig. 1.2c.

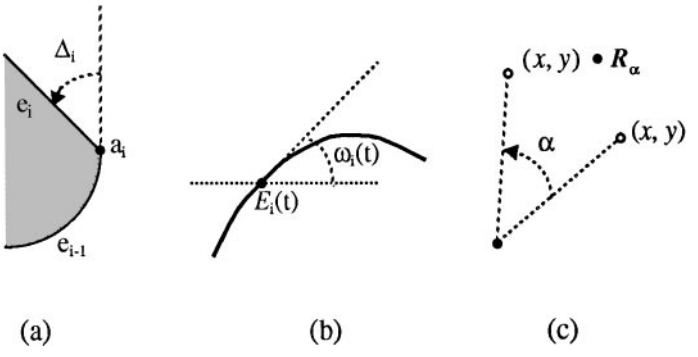


FIG. 1.2. Definitions of Δ_i, $\omega_i(t)$, and $\mathbf{R}_\alpha$.

$\mathbf{T_v}$: The translation operator in the plane, where $\mathbf{v}$ is a vector $(\mathbf{v}_x, \mathbf{v}_y)$. Given a point (x, y), the application of $\mathbf{T_v}$ on it, i.e., $(x, y) \cdot \mathbf{T}_v$, yields a new point $(x', y') = (x + \mathbf{v}_x, y + \mathbf{v}_y)$.

2. CONTACT CONFIGURATIONS OF CONVEX $\mathcal{A}$ AND CONVEX $\mathcal{B}$

This section establishes the foundation of contact configuration space: the construction of $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ when both $\mathcal{A}$ and $\mathcal{B}$ are convex. Because of the convexity of both $\mathcal{A}$ and $\mathcal{B}$, the contact between an instance of $\mathcal{A}$ and $\mathcal{B}$ must fall in one of the three types: an a_i touches an E_j, an e_i touches a b_j, or an e_i touches an E_j. Depending on whether the contact edge e_i is a line segment or an arc, the last two types are further divided into four types. Thus, there are total of five types of contact between $\mathcal{A}$ and $\mathcal{B}$, each giving rise to a unique mathematical representation of faces on $\mathcal{C}_{\mathcal{A},\mathcal{B}}$, as we show next.

Type I. a–E contact. Consider a particular instance $\mathcal{A}(x, y, \theta)$ such that a vertex a_i contacts an edge E_j. Refer to Fig. 2.1 for the geometric entities involved, where τ and τ' are respectively the tangent vectors of edges e_i and e_{i-1} at a_i, v is the angle between the tangent vector of $E_j(t)$ and τ, and L is the length between the reference point $g = (x, y)$ and the vertex a_i.

The following equations are then in order:

$$(x, y) = E_j(t) + (L, 0) \cdot \mathbf{R}_{\omega_j(t)+v+\alpha}; \quad \theta = \omega_j(t) + v + \alpha + \pi - \beta. \tag{1}$$

Since all L, α, and β are constants, the equations above then become functions of t and v only. As all $E_j(t)$, $\omega_j(t)$, and $\mathbf{R}_t$ are smooth functions of a real number t, it turns out that (1) form a C^1 continuous parametric surface in the X–Y–θ space: $\{x(t, v), y(t, v), \theta(t, v): (t, v) \in [0, 1] \times [0, \Delta_i]\}$. In the special case of a linear E_j, the slope function $\omega_j(t)$ degenerates to a constant, and the surface defined by (1) then becomes a ruled surface with all its rulings parallel to the X–Y plane; this conforms to the result in [1].

Type II. e–b contact (l–b and c–b). This corresponds to the contact between an edge e_i and a vertex b_j. Depending on whether e_i is a line segment or an arc, it is further divided into an l–b or c–b contact. Consider first the l–b contact, and let $\mathcal{A}(x, y, \theta)$ be an instance

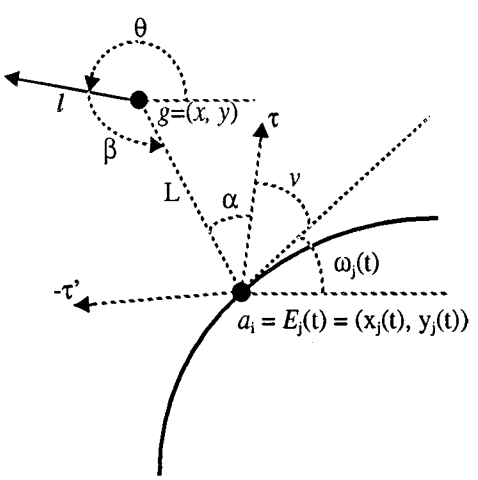


FIG. 2.1. Geometric description of type I contact.

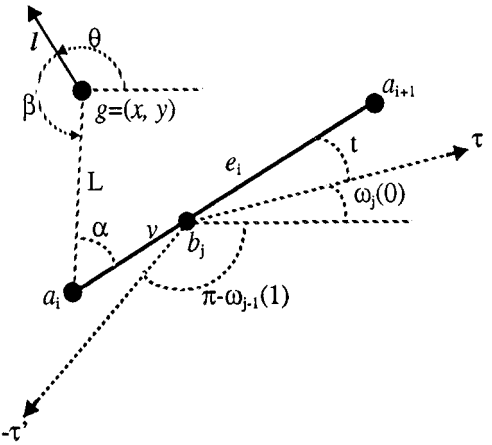


FIG. 2.2. Geometric description of *l*–*b* contact.

that realizes this contact. In Fig. 2.2, τ and τ' are respectively the left and right tangent vectors at b_j, t is the angle measured from τ to e_i, and v is the distance between a_i and b_j. The contact is essentially a "type A" one as defined in [8] if the edges E_j and E_{j-1} are replaced by their tangents τ and $-\tau'$ at b_j. Therefore, the points (x, y, θ) should form a ruled surface in the X–Y–θ space. It is a parametric surface of parameters t and v, as defined by the equations

$$(x, y) = b_j + (L, 0) \cdot \mathbf{R}_{\omega_j(0)+t+\alpha} \cdot \mathbf{T}_{(-v,0)\cdot \mathbf{R}_{\omega_j(0)+t}}; \quad \theta = \omega_j(0) + t + \alpha + \pi - \beta. \quad (2)$$

Note that in the expressions of (2), only t and v are variables, while all the others are constants. The range for (t, v) is $[0, \omega_{j-1}(1) - \omega_j(0)] \times [0, \|a_{i+1} - a_i\|]$, where $\|a_{i+1} - a_i\|$ is the length of edge e_i.

Next, consider the case where e_i is a circular arc. The geometric relationship between an instance $\mathcal{A}(x, y, \theta)$ and $\mathcal{B}$ is manifested in Fig. 2.3. The two control parameters in the figure are t and v, with t being the angle between τ and the tangent line of arc e_i at b_j, and

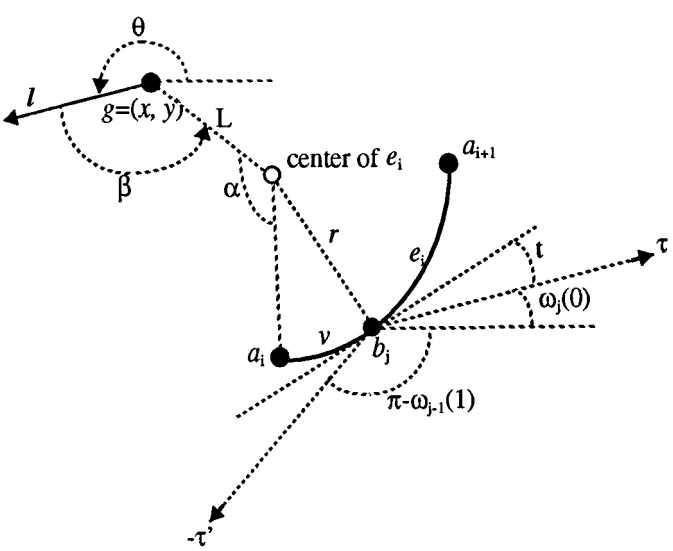


FIG. 2.3. *c*–*b* contact.

v being the arc length of e_i between a_i and b_j. The triplet (x, y, θ) is then related to (t, v) by the equations

$$(x, y) = b_j + (L, 0) \cdot \mathbf{R}_{\omega_j(0)+t+0.5\pi-(v/r)+\pi-\alpha} \cdot \mathbf{T}_{(r,0)\cdot\mathbf{R}_{\omega_j(0)+t+0.5\pi}};$$
$$\theta = \omega_j(0) + t + 2.5\pi - v/r - \alpha - \beta. \tag{3}$$

Similarly to *l–b*, the set of points as described by (3) form a smooth parametric surface $\{x(t, v), y(t, v), \theta(t, v)\}$, with the parameter domain $[0, \omega_{j-1}(1) - \omega_j(0)] \times [0, \|a_{i+1} - a_i\|]$, where $\|a_{i+1} - a_i\|$ this time stands for the arc length of the circular edge e_i. On the other hand, due to the nonlinearity of e_i, the surface is no longer ruled. Instead, it is what we informally call a *circularly ruled surface*. Based on the inversion principle [17], it can be easily shown that if a plane $\theta = \theta_0$ intersects $\{x(t, v), y(t, v), \theta(t, v)\}$, the cross section is always an arc of the same radius r but with varying arc length between zero and $\|a_{i+1} - a_i\|$.

Type III. e–E contact (l–E and c–E). This last type deals with the contact between an edge e_i and an edge E_j, which did not require any consideration for polygonal $\mathcal{A}$ and $\mathcal{B}$. (The contact configurations of two line segments simply degenerate into line segments in the X–Y–θ space.) The *l–E* contact pertains to a linear e_i, whereas the *c–E* contact relates to an arctic e_i. Let (x, y, θ) be a configuration point such that the linear edge e_i touches E_j at a point $E_j(t)$. Obviously, due to the noninterference constraint, e_i must overlap with the tangent vector of E_j at t. Refer to Fig. 2.4. Taking the distance $v = \|a_i - E_j(t)\|$ to be another controlling variable, we have the following equations:

$$(x, y) = E_j(t) + (L, 0) \cdot \mathbf{R}_{\omega_j(t)+\alpha} \cdot \mathbf{T}_{(-v,0)\cdot\mathbf{R}_{\omega_j(t)}}; \quad \theta = \omega_j(t) + \alpha + \pi - \beta. \tag{4}$$

As L, α, and β are all constants, the equations (4) define a parametric surface $\{x(t, v), y(t, v), \theta(t)\}$ with domain $[0, 1] \times [0, \|a_{i+1} - a_i\|]$. Since $\theta(t)$ is unrelated with v, after substituting $\omega_j(t)$ with $\theta - \alpha - \pi + \beta$ in (4), we have $(x(t, v), y(t, v)) = E_j(t) + (L, 0) \cdot \mathbf{R}_{\theta-\pi+\beta} \cdot \mathbf{T}_{(-v,0)\cdot\mathbf{R}_{\theta-\alpha-\pi+\beta}}$, which is a line segment for a fixed t. Therefore, $\{x(t, v), y(t, v), \theta(t)\}$ is a ruled surface with its rulings all parallel to the X–Y plane.

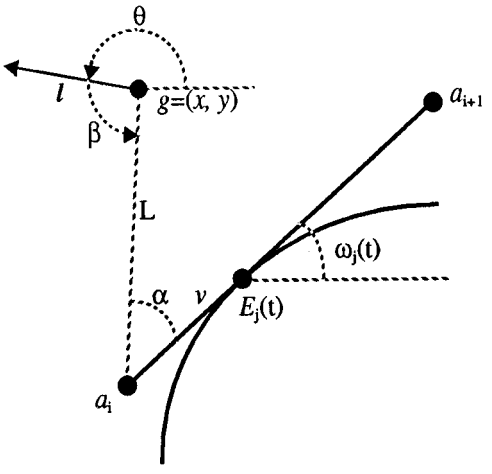


FIG. 2.4. Geometric description of the *l–E* contact.

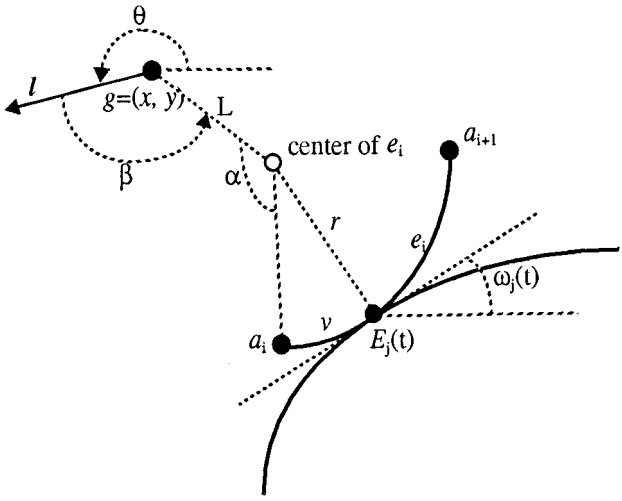


FIG. 2.5. *c–E* contact.

Finally, suppose the edge e_i is an arc of radius r and let (x, y, θ) be a configuration point such that e_i touches E_j at a point $E_j(t)$. Once again, the noninterference requirement dictates that e_i and E_j share a same tangent line at $E_j(t)$. Similarly, let v be the (arc) length of the portion of e_i between a_i and $E_j(t)$. The relationship between (x, y, θ) and t and v, though, is quite different from that of an *l–E* contact, obviously due to the nonlinearity of e_i. It is described by the following equations as shown in Fig. 2.5:

$$\begin{aligned}(x, y) &= E_j(t) + (L, 0) \cdot \mathbf{R}_{\omega_j(t)+0.5\pi-(v/r)+\pi-\alpha} \cdot \mathbf{T}_{(r,0)\cdot\mathbf{R}_{\omega_j(t)+0.5\pi}};\\ \theta &= \omega_j(t) + 2.5\pi - v/r - \alpha - \beta.\end{aligned} \tag{5}$$

The parametric surface $\{x(t, v), y(t, v), \theta(t, v): (t, v) \in [0, 1] \times [0, \|a_{i+1} - a_i\|]\}$ as defined by (5) is closely related to the *offset cylinder* of E_j, i.e., the cylindrical surface with the base curve $\{E_j(t) + (r, 0) \cdot \mathbf{R}_{\omega_j(t)+0.5\pi}: t \in [0, 1]\}$ and the axis being the θ axis. Specifically, the intersection between (5) and a plane $\theta = \theta_0$, if they intersect, is a curve in the form of $\{E_j(t) + (r, 0) \cdot \mathbf{R}_{\omega_j(t)+0.5\pi} + \mathbf{u}(\theta_0): t \in [t_1(\theta_0), t_2(\theta_0)]\}$, where the 2D vector $\mathbf{u}$ and the real numbers t_1, t_2 are C^0 continuous functions of θ. In the extreme case when the reference point g coincides with the center of e_i, $\mathbf{u}$ becomes a null vector function (i.e., with zero length), and, as expected, the surface of (5) turns out to be a connected subset on the offset cylinder. Refer to Fig. 2.6 for two pictorial examples.

A contact configuration, as defined by Eqs. (1) through (5), can thus be represented as $\mathcal{C}|^{\sigma}_{i,j}(t, v)$, where the symbol σ is one of the three types of contact, i.e., *a–E*, *e–b* (for *l–b* and *c–b*), and *e–E* (for *l–E* and *c–E*). For example, $\mathcal{C}|^{a-E}_{i,j}(t, v)$ stands for the type I contact between the vertex a_i and the edge E_j at point $E_j(t)$. Without loss of generality, we can assume the parameter v in (1) through (5) is unitized. There are a total of exactly $m(n-1)$ type I $\mathcal{C}|^{\sigma}_{i,j}(t, v)$ surfaces, mn type II $\mathcal{C}|^{\sigma}_{i,j}(t, v)$ surfaces, and $m(n-1)$ type III $\mathcal{C}|^{\sigma}_{i,j}(t, v)$ surfaces. They constitute the faces of the contact configurations $\mathcal{C}_{\mathcal{A},\mathcal{B}}$. The topology of $\mathcal{C}_{\mathcal{A},\mathcal{B}}$, i.e., the vertices, the edges, and their adjacency relations with the faces, is readily available. For instance, the two *t-edges* of face $\mathcal{C}|^{a-E}_{i,j}(t, v)$ (at $v = 0, 1$), $\mathcal{C}|^{a-E}_{i,j}(t, 0)$ and $\mathcal{C}|^{a-E}_{i,j}(t, 1)$, are exactly shared by the *t*-edges $\mathcal{C}|^{e-E}_{i,j}(t, 0)$ and $\mathcal{C}|^{a-E}_{i-1,j}(t, 1)$, respectively.

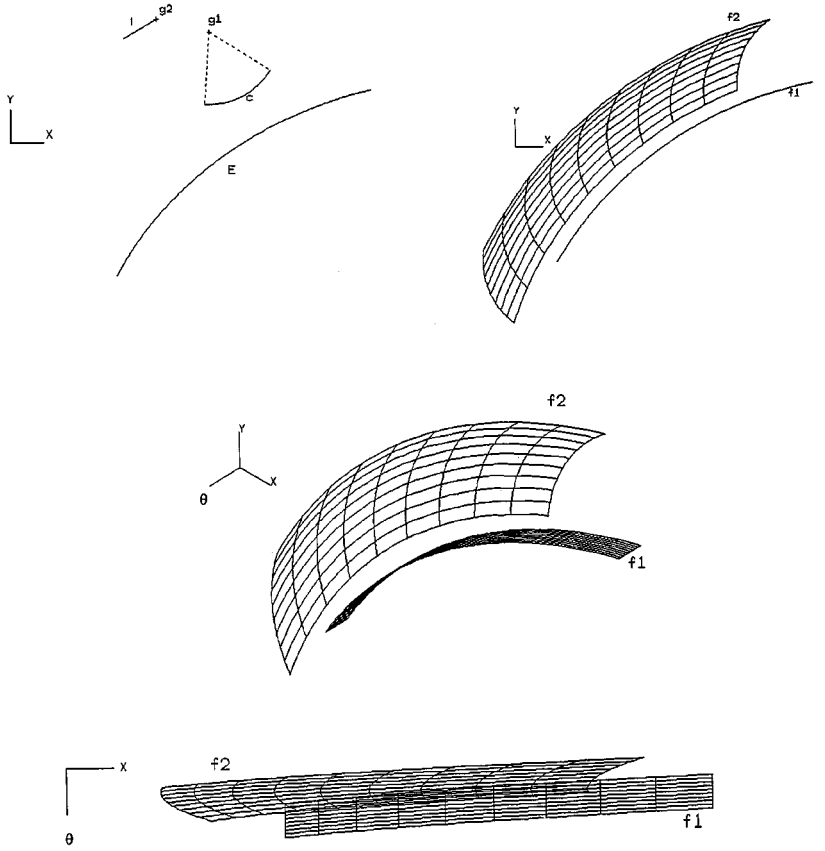


FIG. 2.6. Two examples of c–E contact. f1: c–E contact configurations when g1 (center of arc c) is the reference point of c. f2: c–E contact configurations when g2 is the reference point of c. Both have the same reference vector l.

Figure 2.7 shows such an example of $\mathcal{C}_{\mathcal{A},\mathcal{B}}$. Notice that since a $\mathcal{C}|_{i,j}^{\sigma}(t, v)$ is periodic in θ (of period 2π), only one of its instances is displayed in Fig. 2.7. As revealed in the figure, $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ is not homogeneous. In other words, an edge of a face in $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ may be shared by more than one face. This though can be easily resolved by partitioning the involved faces along certain isoparametric curves.

3. CONTACT CONFIGURATIONS OF A MONOTONE $\mathcal{B}$

Still with respect to a convex $\mathcal{A}$, the $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ of a monotone, but not necessarily convex, $\mathcal{B}$ is naturally built upon the contact configurations of its convex components. Let $\{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_N\}$ be the minimum convex decomposition of $\mathcal{B}$, where each $\mathcal{B}_k$ is a convex chain. Due to its monotonicity, the $\mathcal{C}_{\mathcal{A},\mathcal{B}_k}$ of each $\mathcal{B}_k$ can be viewed as a function of x and θ; let $\mathcal{C}_k(x, \theta)$: $(x, \theta) \in \Omega_k$ denote such a function, $k = 1, 2, \ldots, N$, where each Ω_k is a simple and connected domain in X–θ. We now augment $\mathcal{C}_k$ into a new function $\mathcal{C}_k(x, \theta)$ which will be defined in the domain $\bigcup_{k=1}^{N} \Omega_k$: $\mathcal{C}_k(x, \theta)$ is identical to $\mathcal{C}_k(x, \theta)$ if $(x, \theta) \in \Omega_k$; otherwise it is set to be $-\infty$. We define an upper-enveloping function $\mathcal{C}_{\mathcal{B}}(x, \theta)$ with the domain $\bigcup_{k=1}^{N} \Omega_k$:

$$\mathcal{C}_{\mathcal{B}}(x, \theta) = \text{Maximum}\{\mathcal{C}_k(x, \theta): k = 1, 2, \ldots, N\}. \tag{6}$$

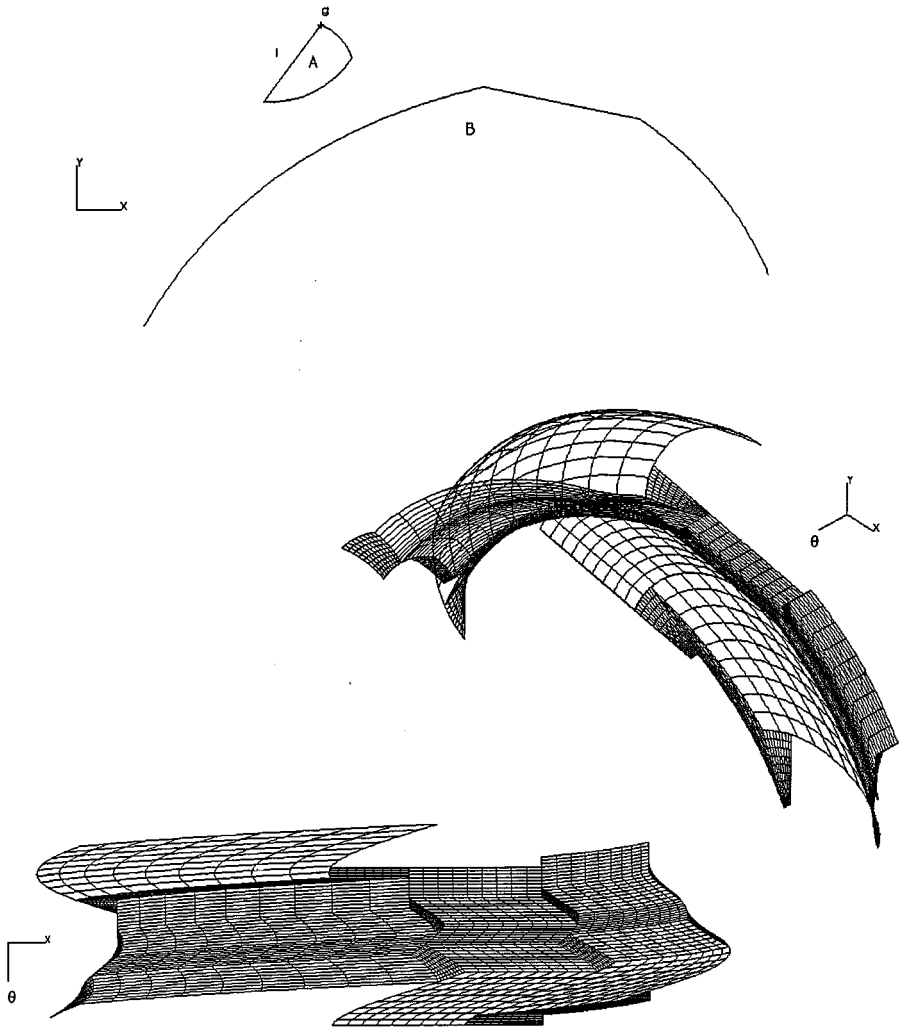


FIG. 2.7. An example of $\mathcal{C}_{\mathcal{A},\mathcal{B}}$.

That $\mathcal{C}_{\mathcal{B}}(x, \theta)$ is indeed the contact configuration space $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ is validated by this observation: because of the monotonicity of $\mathcal{B}$, if an instance $\mathcal{A}(x, y, \theta)$ gouge-freely contacts $\mathcal{B}$, then all the instances $\{\mathcal{A}(x, y + \delta, \theta): \delta > 0\}$ clear $\mathcal{B}$, that is, they can never interfere with the interior $\mathcal{B}_>$. Now that the point (x, y, θ) must be on some $\mathcal{C}_k(x, \theta)$, it follows that $\mathcal{C}_{\mathcal{B}}(x, \theta)$ must be the $\mathcal{C}_{\mathcal{A},\mathcal{B}}$. An illustrative example of $\mathcal{C}_{\mathcal{B}}(x, \theta)$ is shown in Fig. 3.1, where $\mathcal{B}$ is made of two convex subchains $\mathcal{B}_1$ and $\mathcal{B}_2$.

Structurally, $\mathcal{C}_{\mathcal{B}}(x, \theta)$ is described by a set of trimmed faces of the composing $\mathcal{C}_k(x, \theta)$'s. Informally, for a face in $\mathcal{C}_k(x, \theta)$, we first try to intersect it with the faces in all the other $\mathcal{C}_{k'}(x, \theta)$'s ($k' \neq k$). The intersection curves, if any, partition the face into a number of disjoint subpatches which can be easily shown to meet the *congruency* property: a subpatch either entirely belongs to $\mathcal{C}_{\mathcal{B}}(x, \theta)$ or entirely does not belong to $\mathcal{C}_{\mathcal{B}}(x, \theta)$. To compute them, a general algorithm **CC_SURFACE** consisting of three phases—intersection, trimming, and upper-enveloping—is adopted. The input to the algorithm are the N individual contact configuration surfaces $\{\mathcal{C}_k(x, \theta): k = 1, 2, \ldots, N\}$, and the output are the trimmed faces of those $\mathcal{C}_k(x, \theta)$ that make up the $\mathcal{C}_{\mathcal{B}}(x, \theta)$.

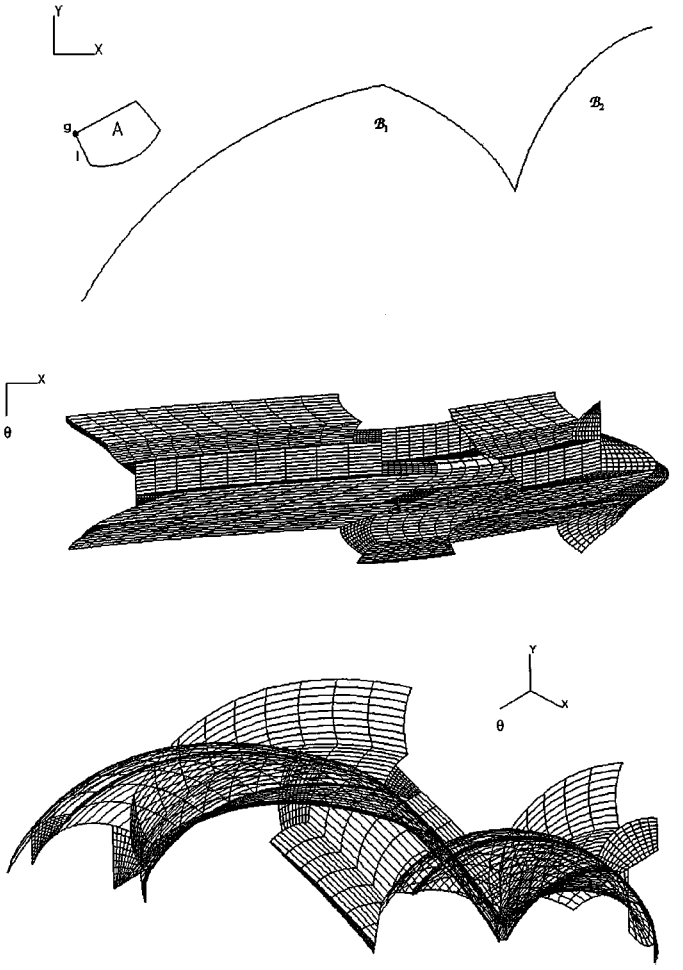


FIG. 3.1. The $\mathcal{C}_{A,\mathcal{B}}$ of a monotone $\mathcal{B}$.

ALGORITHM **CC_SURFACE.**

Phase 1. for every pair of $\langle k, k' \rangle (k \neq k'$, and $k, k' \leq N)$ do
  for every pair of faces $f \in \mathcal{C}_k(x, \theta)$ and $f' \in \mathcal{C}_{k'}(x, \theta)$ do
    if f and f' intersect each other then {
      ∂_f <- $\partial_f \cup (f \cap f')$ /* sets ∂_f and $\partial_{f'}$ are initially empty before Phase 1 */
      $\partial_{f'}$ <- $\partial_{f'} \cup (f \cap f')$
    }

Phase 2. for $k = 1$ to N do /* set Π is initially empty before Phase 2 */
  for every face $f \in \mathcal{C}_k(x, \theta)$ do
    add the subpatches of f induced by the intersection curves in ∂_f into Π

Phase 3. for every trimmed face ϕ in Π do {
  (x, y, θ) <- an arbitrary point in ϕ
  if $y \geq \mathcal{C}_k(x, \theta) : 1 \leq k \leq N$ then
    output ϕ to be a face of $\mathcal{C}_{\mathcal{B}}(x, \theta)$
}

Admittedly, many algorithmic optimizations should be asked for this algorithm, in particular Phase 3. One obvious idea is the *transverseness* of an upper-enveloping function: if a trimmed face $\phi \in C_k(x, \theta)$ belongs to $\mathcal{C}_\mathcal{B}$, then any trimmed face $\phi' \in C_k(x, \theta)$ that is adjacent to ϕ cannot be on $\mathcal{C}_\mathcal{B}$; and consequently, ϕ' does not need to be tested at all in Phase 3. This immediately reduces the amount of computation at Phase 3 by at least half. A faster, but much more complicated, approach such as the topological sweep used in [1] might also be applicable here. While how to design a simple and yet efficient construction algorithm of $\mathcal{C}_\mathcal{B}$ for a curved $\mathcal{B}$ remains to be a challenging problem, it is nevertheless out of the scope of this paper. Instead, we elucidate on how the intersections in Phase 1 are implemented in our system.

Let $\{f(x, \theta)\colon x_{11} \le x \le x_{12}$ and $\theta_{11} \le \theta \le \theta_{12}\}$ be the surface representation of a face $f \in C_k(x, \theta)$, and $\{f'(x, \theta)\colon x_{21} \le x \le x_{22}$ and $\theta_{21} \le \theta \le \theta_{22}\}$ the surface representation of a face $f' \in C_{k'}(x, \theta)$. Depending on the types of f and f', the intersection $f \cap f'$ has different underlying geometric interpretations, as manifested in Fig. 3.2. On the other hand, regardless of the types, a point (x, y, θ) on $f \cap f'$ always designates an instance $\mathcal{A}(x, y, \theta)$ that gouge-freely contacts an edge E on $\mathcal{B}_k$ and an edge E' on $\mathcal{B}_{k'}$ simultaneously. Except for the last case in Fig. 3.2, in which edges E and E' "support" the same linear edge of $\mathcal{A}$, which can be handled as a special case in the intersection algorithm, due to the convexity of $\mathcal{A}$, E, and E', for a given orientation θ, there is at most one instance of $\mathcal{A}(\theta)$ that can gouge-freely contact both E and E' at the same time. Therefore, the intersection $f \cap f'$ is a continuous single curve which is strictly monotone with respect to θ (i.e., its intersection with any plane $\theta = \theta_0$ is at most one point). In light of this monotonicity, a binary-tracing algorithm is devised to compute the intersection $f \cap f'$. Let (x_1, y_1, θ_1) and (x_2, y_2, θ_2) be the two end points of $f \cap f'$, which are obtained by intersecting the four boundary edges

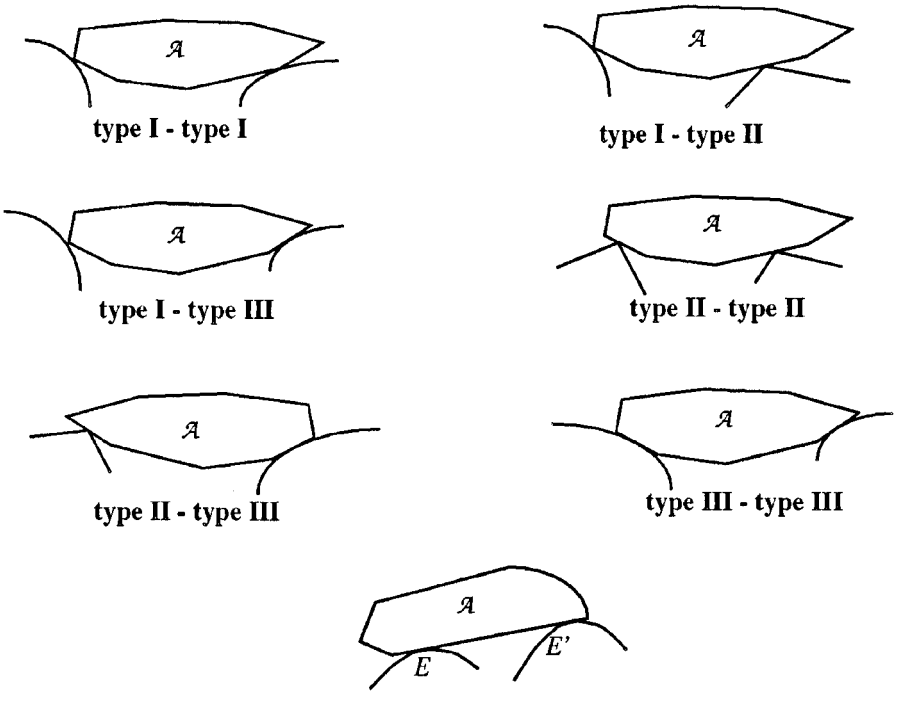


FIG. 3.2. Different configurations of contact between $\mathcal{A}$ and two edges E and E'.

of f with f' and vice versa, by means of a general Newton–Raphson method and taking advantage of the fact that $f \cap f'$ is monotone. The recursive procedure **Intersect_Interval** generates an ordered list of points that represent $f \cap f'$.

Intersect_Interval ($f(x,\theta)$, $f'(x,\theta)$, (x_1,θ_1), (x_2,θ_2))
begin
    if **stop_satisfy** ($f(x,\theta)$, $f'(x,\theta)$, (x_1,θ_1), (x_2,θ_2)) then
        output $(x_1,\ f(x_1,\theta_1),\theta_1)$ and $(x_2,\ f(x_2,\theta_2),\theta_2)$
    else {
        θ_m <- $(\theta_1+\theta_2)/2$
        (x_m, y_m, θ_m) <- $f(x,\theta_m) \cap f'(x,\theta_m)$
        Intersect_Interval ($f(x,\theta)$, $f'(x,\theta)$, (x_1,θ_1), (x_m,θ_m))
        Intersect_Interval ($f(x,\theta)$, $f'(x,\theta)$, (x_m,θ_m), (x_2,θ_2))
    }
end

The procedure **Intersect_Interval** stops recursion when the condition **stop_satisfy** is met. Either of the following two criteria will make **stop_satisfy** true:

(1) The 3D distance between the two points $(x_1,\ f(x_1,\theta_1),\theta_1)$ and $(x_2,\ f(x_2,\theta_2),\theta_2)$ is less than a preset tolerance value; or

(2) The minima of the radii of curvature of $f(x,\theta)$ and $f'(x,\theta)$ at $(x_1,\ f(x_1,\theta_1),\theta_1)$, say r', and at $(x_2,\ f(x_2,\theta_2),\theta_2)$, say r'', are identical (within a preset tolerance), and the ratio of r' over the 3D distance between the two points is greater than some predetermined number δ.

The first criterion ensures that the generated points are close enough to accurately represent the intersection curve. The second one, on the other hand, helps reduce the generation of unnecessary or redundant intersection points. In many occasions, the portion on the intersection curve between the two points may be closely represented by a line segment or an arc. The threshold number δ is empirical. In general, the larger δ is, the more points will be generated.

$f(x,\theta_m) \cap f'(x,\theta_m)$ is a single intersection point between two convex curves in the X–Y plane. According to the analysis of the three types of contact conducted in the previous section, $f(x,\theta_m)$ can only be one of: (1) a line segment, (2) an arc, (3) a portion of a translated edge E of $\mathcal{B}_k$, or (4) a portion of an offset of a translated edge E. The same argument also holds for $f'(x,\theta_m)$. Therefore, this intersection can be computed either algebraically if the corresponding edges on $\mathcal{B}_k$ and $\mathcal{B}_{k'}$ are of lower order degrees, e.g., cubic Bezier curves, or otherwise numerically, which should converge quickly because of the convexity of both edges.

When all $\mathcal{A}$, $\mathcal{B}_k$, and $\mathcal{B}_{k'}$ are polygonal, the intersection between the two faces f and f' becomes simpler. It is now an algebraic/sinusoid function $\mathcal{F}(t)$: $t_1 \le t \le t_2$, where $\mathcal{F}(t)$ takes in one of three types of algebraic and/or sinusoid expressions, depending on the types of f and f', i.e., type I vs type I, type I vs type II, and type II vs type II. (Note that type III faces now degenerate into line segments.) In the Appendix, a formal proof is given for the three expressions. The implication of this is important. Rather than calling the numerical routine **Intersect_Interval** for intersecting f and f', the only thing we need to do is to identify the function $\mathcal{F}(t)$. Our tests results have shown that, for polygonal $\mathcal{A}$ and $\mathcal{B}$, using the function $\mathcal{F}(t)$ greatly reduces the computational time, as compared to the undiscriminating calling of **Intersect_Interval**.

4. CASE OF A CONCAVE $\mathcal{A}$

Finally, we give a general account of how the $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ can be computed for a concave $\mathcal{A}$.

Despite the nonconvexity of $\mathcal{A}$, the idea of the three-phase algorithm—intersection, trimming, and upper-enveloping—still applies, as it is not hard to prove that $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ is still monotone with respect to the X–θ plane. Let $\langle \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_M \rangle$ be a minimum convex decomposition of $\mathcal{A}$, where each $\mathcal{A}_i$ is a convex generalized polygon. In addition, a reference point and vector are chosen which are *common* to all the $\mathcal{A}_i$'s. Let $\mathcal{C}_{ij}(x, \theta)$: $(x, \theta) \in \Omega_{i,j}$ denote the contact configurations of $\mathcal{A}_i$ and $\mathcal{B}_j$ only, i.e., $\mathcal{C}_{\mathcal{A}_i,\mathcal{B}_j}$. Again, because of its monotonicity, the contact configurations $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ is the upper-enveloping function

$$\mathcal{C}_{\mathcal{B}}(x, \theta) = \text{Maximum}\{\mathcal{C}_{i,j}(x, \theta): i = 1, 2, \ldots, N,\ j = 1, 2, \ldots, M\}, \tag{7}$$

where $\mathcal{C}_{i,j}(x, \theta)$ is the augmented $\mathcal{C}_{i,j}(x, \theta)$ defined in the entire domain $\bigcup_{i,j=1}^{N,M} \Omega_{i,j}$.

Figure 4.1 depicts the θ-slice ($0 \le \theta \le 2\pi$) representation of a $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ for a stair-like $\mathcal{A}$. (A θ-slice at $\theta = \theta_0$ is the planar curve $\{(x, \mathcal{C}_{\mathcal{B}}(x, \theta_0), \theta_0)\colon\ x \in [x_1, x_2]\}$, where x_1 and x_2 are the two extreme x-abscissa of the domain $\bigcup_{i,j=1}^{N,M} \Omega_{i,j}$ at $\theta = \theta_0$.) A caution, however, must be emphasized when using the algorithm **CC_SURFACE**. When $\mathcal{A}$ is concave, the intersection between two faces f and f' is no longer guaranteed to be monotone with respect to θ, nor

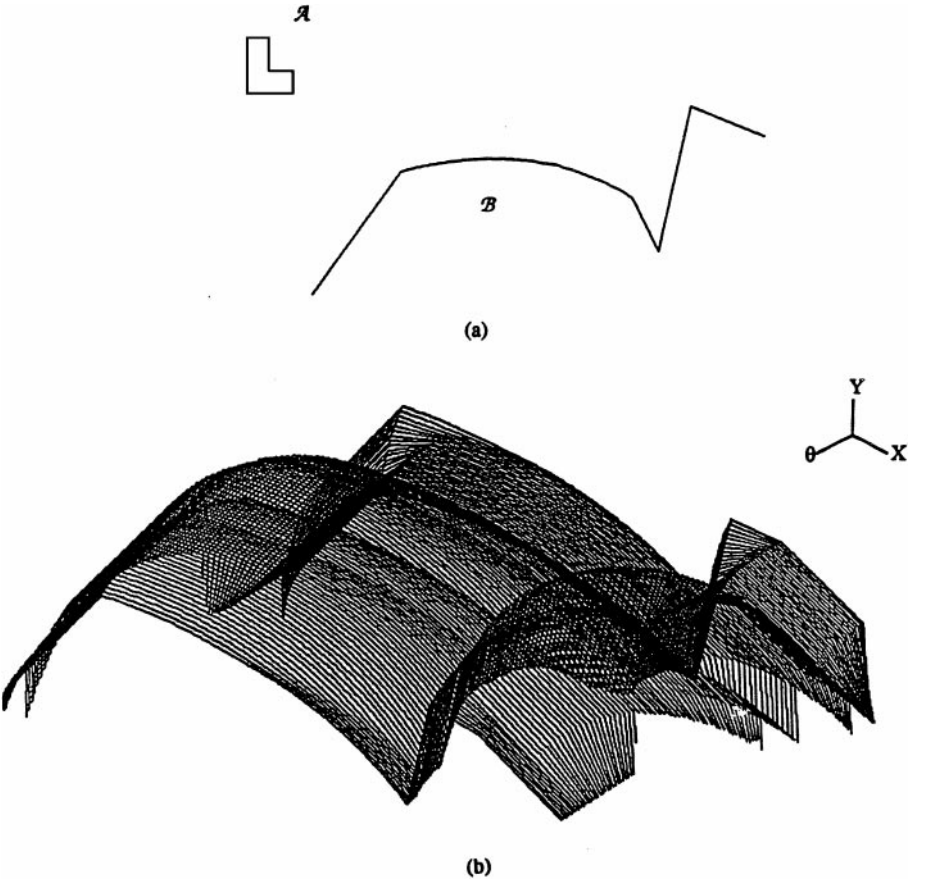


FIG. 4.1. The θ-slice representaion of the $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ of a concave $\mathcal{A}$.

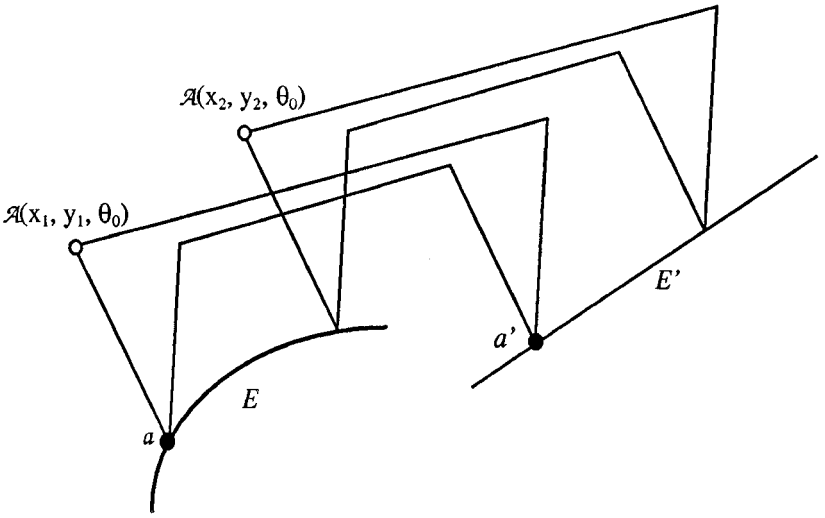


FIG. 4.2. Nonmonotonocity of critical curves due to nonconvexity of $\mathcal{A}$.

is it always a single continuous piece. In the example shown in Fig. 4.2, at the two indicated configurations (x_1, y_1, θ_0) and (x_2, y_2, θ_0), the vertex a (a' respectively) of $\mathcal{A}$ contacts the edge E(E' respectively) of $\mathcal{B}$. The intersection between the face of a–E contact and the face of a'–E' contact (both are of type I) is not monotone with respect to the θ-axis, since the two intersection points (x_1, y_1, θ_0) and (x_2, y_2, θ_0) have the same θ value θ_0. Consequently, the procedure **Intersect_Interval** cannot be used here. Instead, a general surface–surface intersection algorithm is needed to compute the critical curves. In the simplest case, though, i.e., when both $\mathcal{A}$ and $\mathcal{B}$ are polygonal, the procedure **Intersect_Interval** is still valid, since it is easily seen that the situation in Fig. 4.2 cannot exist if both E and E' are linear.

5. AN APPLICATION EXAMPLE OF $\mathcal{C}_{\mathcal{A},\mathcal{B}}$

We next give an example of the applications of $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ in mill-turn machining. In this application, $\mathcal{B}$ is the profile curve that defines the contour of the part that is symmetric to the X-axis (the rotational symmetry line), and the cutter can be modeled as a generalized polygon $\mathcal{A}$, whose orientation is limited to a range $\gamma \subset [0, 2\pi]$. See the example shown in Fig. 1.1. A point p on $\mathcal{B}$ is said to be *machinable* (i.e., *reachable*) by the cutter $\mathcal{A}$ if there exists at least one configuration (x_0, y_0, θ_0) $(\theta_0 \in \gamma)$ such that $\mathcal{A}(x_0, y_0, \theta_0)$ gouge-freely contacts $\mathcal{B}$ at p. Let $\mathcal{R}_{\mathcal{A},\mathcal{B}}(\theta)$ denote the set of points on $\mathcal{B}$ that are reachable from instances of $\mathcal{A}$ with the orientation θ. Each orientation θ represents a distinct *set-up* of the cutter which, due to the remounting and recalibrating of the cutter required, should be limited to as few as possible. An important task thus is: find a minimum set of orientations $\{\theta_1, \theta_2, \ldots, \theta_l\} \in \gamma$ such that $\bigcup_{i=1}^{l} \mathcal{R}_{\mathcal{A},\mathcal{B}}(\theta_i) = \bigcup_{\theta\in\gamma} \mathcal{R}_{\mathcal{A},\mathcal{B}}(\theta)$.

In reality, besides the range $\gamma \subset [0, 2\pi]$, there is an additional constraint put on the machining: only certain edges of $\mathcal{A}$ will physically contribute to the machining of $\mathcal{B}$, e.g., the arc of $\mathcal{A}$ in Fig. 1.1. These edges will be referred to as the *cutting edges* of $\mathcal{A}$.

Now consider the contact between an edge E_j of $\mathcal{B}$ and the cutting edges of $\mathcal{A}$. The corresponding contact configurations are uniquely represented by a subset of faces on $\mathcal{C}_{\mathcal{A},\mathcal{B}}$,

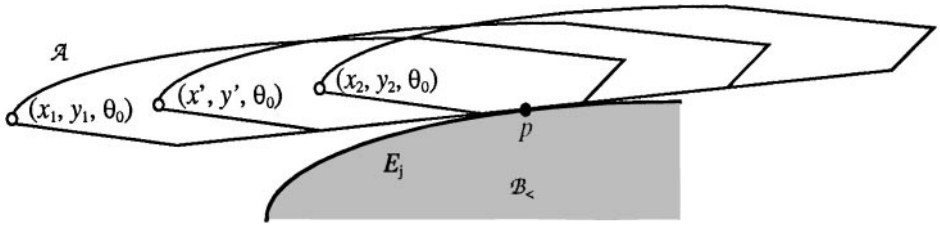


FIG. 5.1. Continuous contact between a linear edge of $\mathcal{A}$ and an edge E_j of $\mathcal{B}$.

say $\{f_1, f_2, \ldots, f_k\}$. Each f_i is a (trimmed) parametric surface of type III that describes the contact between a cutting edge and a portion of edge E_j. The boundary ∂f_i of a face f_i is made of natural boundary edges as defined in equations (4) and (5) and/or some trimming curves. Referring to equations (4) and (5), we notice that for a point (x, y, θ) in ∂f_i, there is one and only one contact point on E_j. Consequently, ∂f_i designates a series of contact points on E_j. Because of the continuity of ∂f_i, this series of points form a continuous and closed curve in the X–Y–θ space, whose projection in the X–θ plane bounds a compact region, to be referred to as $\chi(f_i)$. We define the *contact map* of edge E_j to be the union $\Gamma(E_j) = \bigcup_{i=1}^{k} \chi(f_i)$. From the computational point of view, it is worth mentioning that, although in general two different $\chi(f_i)$ and $\chi(f_{i'})(i \neq i')$ may intersect each other, they are always disjoint when $\mathcal{A}$ is convex. For if two (convex) instances $\mathcal{A}(x_1, y_1, \theta_0)$ and $\mathcal{A}(x_2, y_2, \theta_0)$ both contact the same point p, the only possibility is that a linear edge of $\mathcal{A}$ contacts p, as shown in Fig. 5.1. However, this would imply that every point (x', y', θ_0) lying on the line segment between the two points (x_1, y_1, θ_0) and (x_2, y_2, θ_0) must belong to $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ too, due to the monotonicity of $\mathcal{B}$. Therefore, (x_1, y_1, θ_0) and (x_2, y_2, θ_0) are connected on $\mathcal{C}_{\mathcal{A},\mathcal{B}}$, and hence belong to a same face f_i.

The contact map of $\mathcal{B}$ is then defined as the union $\Gamma(\mathcal{B}) = \bigcup_{j=1}^{n-1} \Gamma(E_j)$. Notice that every pair of $\Gamma(E_j)$ and $\Gamma(E_{j'})$ $(j \neq j')$ are always disjoint, regardless of the convexity of $\mathcal{A}$. As an illustrative example, Fig. 5.2 displays the $\Gamma(\mathcal{B})$ of a convex polygon with a filet arc. For simplicity, all the five edges (including the filet arc) of $\mathcal{A}$ in Fig. 5.2 are taken as the cutting edges, and the range γ is $[0, 2\pi]$.

Let $\alpha(\theta_0)$ denote the projection on the X-axis of the intersection between the line $\theta = \theta_0$ and the contact map $\Gamma(\mathcal{B})$. The minimization problem then equals to finding a minimum set of real numbers $\{\theta_1, \theta_2, \ldots, \theta_l\} \in \gamma$ such that the union $\bigcup_{i=1}^{l} \alpha(\theta_i)$ is equal to the projection of $\Gamma(\mathcal{B})$ on the X-axis. This is apparently an NP-complete problem [5]. Furthermore, the nonlinearity of the edges on $\Gamma(\mathcal{B})$ makes it more ominous of finding an exact optimal solution. As an alternative, we propose a "greedy" approach [16]. Basically, a line $\theta = \theta_1$ is initially found that realizes the maximum $\alpha(\theta)$. In the example of Fig. 5.2, this is a line $\theta = \theta_l = 1.785$ (102.24°). The contact map $\Gamma(\mathcal{B})$ then is intersected with the strips corresponding to the intervals $\{[x_1, x_n] - \alpha(\theta_1)\}$, resulting in a "shrunk" contact graph $\Gamma_1(\mathcal{B})$, as shown in Fig. 5.3. This intersecting–shrinking process is then recursively continued until the final $\Gamma_l(\mathcal{B})$ becomes empty, generating a series of $\{\theta_1, \theta_2, \ldots, \theta_l\}$, e.g., $\{1.785, 1.0\}$ in Fig. 5.3. While this series may be longer than the theoretical exact solution, it should suffice in practice. To find a maximum $\alpha(\theta)$, the well-known plane-sweep method [11] is used, in which a sweep line parallel to the X-axis scans along the θ-axis, and the events are those occurring when the sweep line encounters the end points of the edges representing the boundary of $\Gamma_l(\mathcal{B})$.

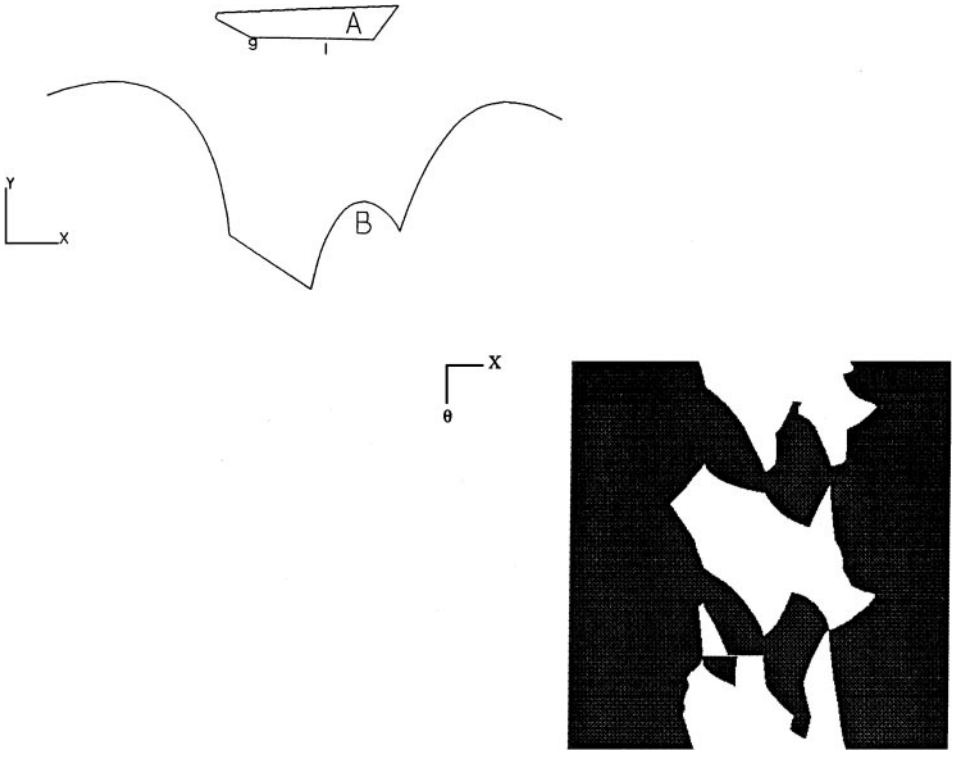


FIG. 5.2. An example of contact map $\Gamma(\mathcal{B})$.

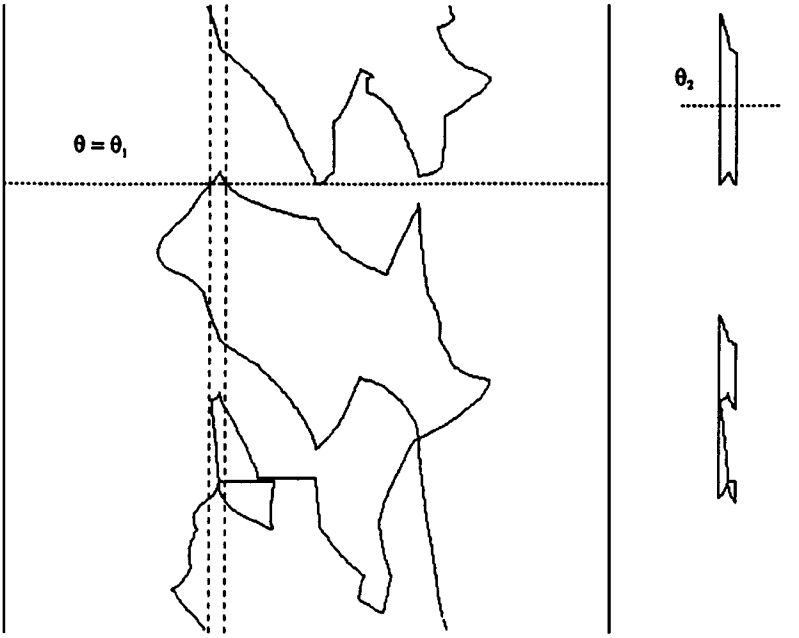


FIG. 5.3. The intersecting–shrinking solution to the minimization problem.

6. SUMMARY

The primary goal of this paper is to establish an exact mathematical description of the gouge-free contact between a concave generalized polygon $\mathcal{A}$ (with line segments and arcs) and an oriented monotone chain $\mathcal{B}$ of parametric curved edges. We show that the corresponding contact configuration space is a surface defined by a set of trimmed parametric surfaces (the faces) and give their exact mathematical representations. Various useful geometric properties of these faces are then analyzed. A practical algorithm is proposed for computing the intersection of the faces, which contribute to the critical curves on the contact configuration surface. In addition, exact algebraic/sinusoid parametric expressions are given for the critical curves in the special case when both $\mathcal{A}$ and $\mathcal{B}$ are polygonal. An application in mill-turn machining is then presented to illustrate the usefulness of our results.

We have implemented the algorithms, using C language on a PC with a modest configuration of 300 MHZ PII CPU and 64 MB RAM. In the tests, for an arbitrary concave $\mathcal{B}$ of between 30 and 40 cubic Bezier curves and a convex $\mathcal{A}$ of 20 or so arc/line edges—a reasonable input size in the mill-turn machining application—the configuration space $\mathcal{C}_{\mathcal{A},\mathcal{B}}$ is computed quickly, usually in less than 15 seconds (excluding data input drivers and graphics display). When $\mathcal{A}$ is concave but the number M of its convex components is small, there is no obvious run-time increase. When M is relatively large, say over 20, significant run-time increase is observed. This is in major part due to that a general numerical algorithm has to be used to compute an intersection $f \cap f'$, rather than the fast binary-tracing procedure **Intersect_Interval**. The numerical algorithm we used is rather primitive and converges slowly, and it is believed that a better one will greatly help reduce the run-time.

Future topics for research remain. Among them one is to remove the local convexity requirement on the edges of $\mathcal{B}$, or at least to allow some special types of curves (such as conics), if a general solution seems to be too ominous. Currently in our system, a concave curved edge has to be approximated by line segments which tremendously increase the computing time and numerical sensitivity. Another issue is to see if faster construction algorithms for the contact configurations space can be devised. Finally and not least, how to extend our results to an $\mathcal{A}$ of general parametric curved edges posts to be a challenging problem.

APPENDIX

Let $\mathcal{B}_i$ and $\mathcal{B}_j$ be two convex sub-chains on $\mathcal{B}$ and consider the intersection between two faces $f \in \mathcal{C}_{\mathcal{A},\mathcal{B}_i}$, and $f' \in \mathcal{C}_{\mathcal{A},\mathcal{B}_j}$. We show that if the associated geometric entities of f and f' on $\mathcal{A}$ and $\mathcal{B}$ are vertices and line segments only, the intersection $f \cap f'$ is a single simple curve in the X–Y–θ space that can always be described by a closed parametric form.

Several tool functions are first needed. Refer to Fig. A.1. Given a rigid triangle with sides of lengths l_1, l_2, and l_3, and the X–Y coordinates of two vertices (x_1, y_1) and (x_2, y_2) known, the coordinates of the third vertex, (x_3, y_3), is an algebraic/sinusoid function of $\{x_1, y_1, x_2, y_2, l_1, l_2, l_3\}$. So let this function be denoted as *tri_coords*, i.e., $(x_3, y_3) = tri\_coords(x_1, y_1, x_2, y_2, l_1, l_2, l_3)$. Also, the reference angle θ, measured from the $+X$ axis to the vector $(x_3, y_3) \rightarrow (x_1, y_1)$, can be algebraically expressed in terms of $\{x_2, y_2, x_3, y_3, \alpha\}$. Let *tri_angle* denote this expression, i.e., $\theta = tri\_angle(x_2, y_2, x_3, y_3, \alpha)$. Finally, we will use $quad\_sol(a, b, c)$ to represent the valid one of the two solutions to the real number quadratic equation $ax^2 + bx + c = 0$. (The validity of *quad_sol* will become clear later in the context.)

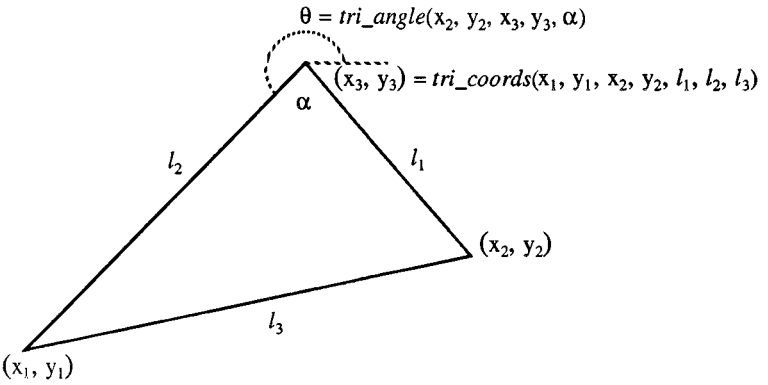


FIG. A.1. Algebraic expressions in a triangle.

The parametric expression of $f \cap f'$ takes in one of three different forms, depending on the types of f and f'.

Type I–Type I. This corresponds to the situation where two vertices of $\mathcal{A}$ slide simultaneously along two different linear edges of $\mathcal{B}$. Without loss of generality, assume the two contacting edges of $\mathcal{B}$ are placed as shown in Fig. A.2. We take the Y coordinate of the vertex v_a as the control parameter t. The following derivations are in order:

(a) $x_b^2 + (y_b - t)^2 = l_1^2$

(b) $y_b = cx_b$

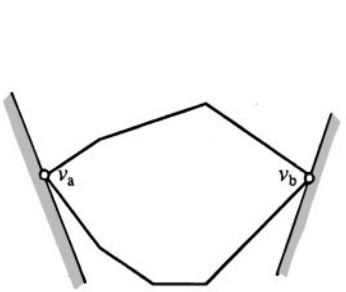
(c) $(1 + c^2)x_b^2 + (-2ct)x_b + \left(t^2 - l_1^2\right) = 0$ (because of (a) and (b))

(d) $x_b = \mathit{quad\_sol}\left(1 + c^2, -2ct, t^2 - l_1^2\right)$

(e) $y_b = cx_b$

(f) $(x_1, y_1) = \mathit{tri\_coords}(0, t, x_b, y_b, l_a, l_b, l_1)$

(g) $\theta = \mathit{tri\_angle}(0, t, x_1, y_1, \alpha)$.



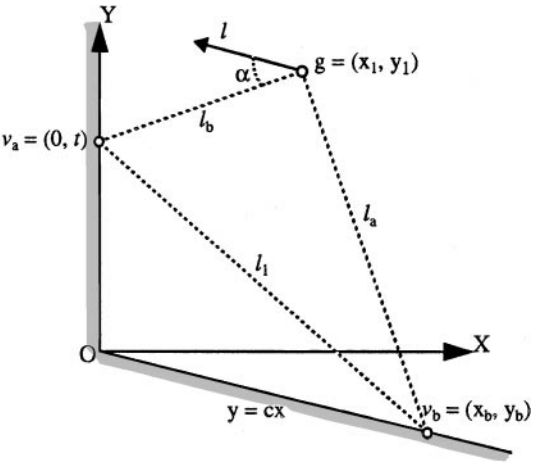


FIG. A.2. Type I–Type I.

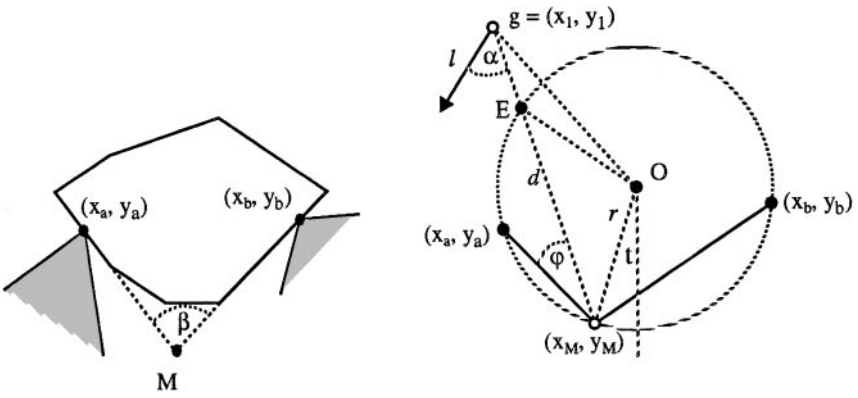


FIG. A.3. Type II–Type II.

Thus, $\mathcal{F}(t) = (x_1, y_1, \theta)$ is a function of a single parameter t. In particular, if the two contact edges are orthogonal to each other, i.e., the vertex v_b moves along the X axis, it can be shown that g traces an ellipse in the X–Y plane. (Refer to [3] for details.)

Type II–Type II. This time, two different linear edges of $\mathcal{A}$ are constrained by two vertices of $\mathcal{B}$. That is, $\mathcal{A}$ can move and rotate, but these two edges have to keep constant contact with the two vertices.

Refer to Fig. A.3. Because the angle β at point M is a constant, point M must trace a circle. Without loss of generality, assume that the center of this circle is the origin O. By designating the angle between the $-Y$ axis and the vector $\mathrm{O} \rightarrow (x_\mathrm{M}, y_\mathrm{M})$ as the control parameter t, we have the following derivations:

(a) $(x_\mathrm{M}, y_\mathrm{M}) = (-r \times \sin(t), -r \times \cos(t))$

(b) $l_1 = \|(x_\mathrm{M}, y_\mathrm{M}) - (x_\mathrm{a}, y_\mathrm{a})\| = \sqrt{(x_\mathrm{M} - x_\mathrm{a})^2 + (y_\mathrm{M} - y_\mathrm{a})^2}$

(c) $l_\mathrm{M} = \|(x_1, y_1) - (x_\mathrm{a}, y_\mathrm{a})\| = \sqrt{d^2 + l_1^2 - (2 \cdot d \cdot l_1 \cdot \cos\varphi)}$

(d) $(\mathrm{x}_1, \mathrm{y}_1) = \mathit{tri\_coords}(x_\mathrm{a}, y_\mathrm{a}, x_\mathrm{M}, y_\mathrm{M}, d, l_\mathrm{M}, l_1)$

(e) $\theta = \mathit{tri\_angle}(x_\mathrm{M}, y_\mathrm{M}, x_1, y_1, \alpha)$.

Note that the symbol "d" above represents a constant; it is the distance between the reference point g and point M. So is the angle φ. Since $(x_\mathrm{a}, y_\mathrm{a})$ is a fixed point, once again (x_1, y_1, θ) is a function of a single parameter t. Actually, the path of (x_1, y_1) can be analytically expressed. It can be shown (see [3]) that the intersection point E in Fig. A.3 is a fixed point, regardless of the angle t. Taking E as the origin, the polar coordinates ρ and ϕ of g are then related by the equation

$$\rho = 2r \cdot \cos\phi + d,$$

where ρ is the distance measured from E to g and ϕ is the angle between the two vectors $\mathrm{E} \rightarrow \mathrm{O}$ and $\mathrm{E} \rightarrow g$. This is the representation of a Pascal or snail-like curve [3].

Type I–Type II. This last type corresponds to the scenario when a linear edge and a vertex of $\mathcal{A}$ are constrained by a vertex and a linear edge of $\mathcal{B}$ respectively, as demonstrated in Fig. A.4. In terms of motion geometry mechanisms, g traces a *Conchoidal* motion

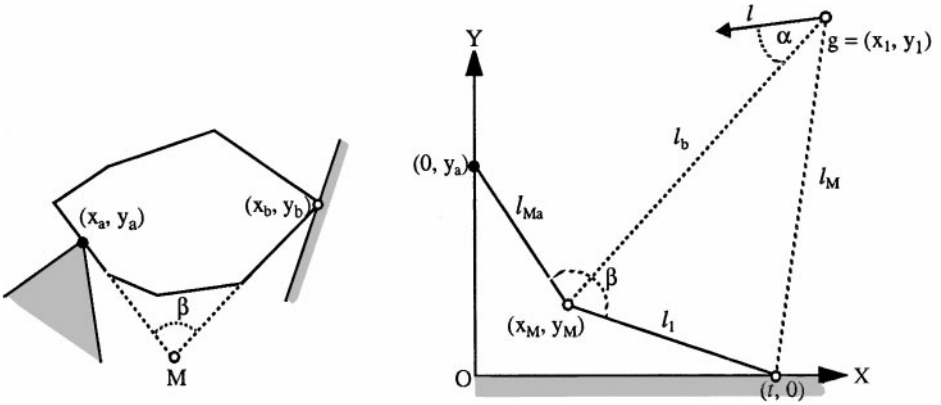


FIG. A.4. Type I–Type II.

curve [3, 7]. Assuming the X–Y coordinates system shown in the figure and taking the X coordinate of the sliding vertex of $\mathcal{A}$ as the control parameter t, we have

(a) $l_{ab} = \|(x_b, y_b) - (x_a, y_a)\| = \sqrt{t^2 + y_a^2}$

(b) $l_{Ma} = \sqrt{x_M^2 + (y_M - y_a)^2}$

$(x_M - t)^2 + y_M^2 = l_1^2$

$l_{ab} = \sqrt{l_{Ma}^2 + l_1^2 - (2 \cdot l_{Ma} \cdot l_1 \cdot \cos\beta)} = \sqrt{t^2 + y_a^2}$ (after substituting (a))

(c) $(x_M, y_M) =$ the unique valid solution to the three equations in (b)

(d) $(x_1, y_1) = tri\_coords(x_M, y_M, t, 0, l_M, l_b, l_1)$

(e) $\theta = tri\_angle(x_M, y_M, x_1, y_1, \alpha)$.

Notice that, given the physical geometry shown in the figure, there is one and only one valid solution in (b) for (x_M, y_M). As l_1, β, and y_a are all constants, the solution for (x_M, y_M) should be a function of the sole variable t. So should (x_1, y_1), since both l_M and l_b are constants too. Therefore, (x_1, y_1, θ) is a function of t.

REFERENCES

1. F. Avnaim, J. D. Boissonnat, and B. Faverjon, A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles, in *Proceedings of IEEE Conference on Robotics and Automation, 1988*, pp. 1656–1661.
2. R. A. Brooks, Solving the find-path problem by good representation of free space, *IEEE Trans. Systems Man Cybernet.* **SMC-13**, 1983, 190–197.
3. E. A. Dijksman, *Motion Geometry of Mechanisms*, Cambridge Univ. Press, Cambridge, UK, 1976.
4. I. Drori, L. Joskowicz, and E. Sacks, Contact analysis of spatial fixed-axes pairs using configuration spaces, in *Proceedings of IEEE Conference on Robotics and Automation, 1999*, pp. 578–584.
5. M. R. Garey and D. S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.

6. K. Kedem and M. Sharir, *An Efficient Motion Planning Algorithm for a Convex Polygonal Object in 2-Dimensional Polygonal Space*, Tech. Report 253, Comp. Sci. Dept., Courant Institute, 1986.
7. J. C. Latombe, *Robot Motion Planning*, Kluwer Academic, Dordrecht/Norwell, MA, 1993.
8. T. Lozona-Perez, Spatial planning: A configuration space approach, *IEEE Trans. Comput.* **C-32**(2), 1983, 108–120.
9. T. Lozona-Perez and M. Wesley, An algorithm for planning collision-free paths among polyhedral obstacles, *Comm. ACM* **22**, 1979, 560–570.
10. T. Lozona-Perez and R. A. Brooks, A subdivision algorithm in configuration space for findpath with rotation, *IEEE Trans. Systems Man Cybernet.* **SMC-15**(2), 1985, 224–233.
11. J. O'Rourke, *Computational Geometry in C*, Cambridge Univ. Press, Cambridge, MA, 1993.
12. J. H. Reif, Complexity of the mover's problem and generalizations, in *Proceedings 20th IEEE Symposium on Foundations of Computer Science, 1979*, pp. 421–427.
13. J. T. Schwartz and M. Sharir, On the piano mover's problem. I. The case of a two dimensional rigid polygonal body moving amidst polygonal barriers, *Comm. Pure Appl. Math.* **36**, 1983, 345–398.
14. J. T. Schwartz and M. Sharir, On the piano mover's problem. II. General techniques for computing topological properties of real algebraic manifolds, *Adv. Appl. Math.* **4**, 1983, 298–351.
15. J. T. Schwartz and M. Sharir, A survey of motion planning and related geometric algorithms, in *Geometric Reasoning* (D. Kapur and J. L. Mundy, Eds.), pp. 158–169, MIT Press, Cambridge, MA.
16. K. Tang, S. Chou, and L. Chen, Optimal workpiece Setups for 4-axis NC machining, *Comput. Industry* **37**(1), 1998, 27–41.
17. K. Tang and Y. Dayan, Offsetting surface boundaries and 3-axis gouge-free surface machining, *Comput. Aided Design* **27**(12), 1995, 915–927
18. S. Udupa, *Collision Detection and Avoidance in Computer Controlled Manipulators*, Ph.D. dissertation, Dept. of Electrical Engineering, California Institute of Technology, 1977.