Critical Curves in the Piano Movers' Algorithm

Ron Wein

March 23, 2005

Abstract

We summarize the Piano Movers' Algorithm for planning a collision-free motion path for a translating and rotating segment among polygonal obstacles and for a translating and rotating polygon among polygonal obstacles. We give a special attention to the analysis of the algebraic curves that appear in the algorithm, and discuss the possibilities of giving a robust software implementation for the algorithm by constructing an arrangement of these curves using the existing software libraries for exact computation.

1 Introduction

In the early 1980's, Schwartz and Sharir published a series of articles discussing complete solutions to several motion-planning problems. In the first article [12], they treated the problem of a moving (translating and rotating) line segment (so-called a *"ladder"*) amidst polygonal obstacles, and of a polygonal robot amidst polygonal obstacles — both problems having three degrees of motion freedom. The "ladder" case is also discussed by Latombe [9] in a somewhat more intuitive manner.

The nice property of the piano-movers' algorithms is that it enables solving the motion-planning problem using only planar geometry, without the need to perform computations in the three-dimensional configuration space. The availability of software tools for handling planar geometry makes the algorithms more feasible to implement, in comparison to algorithms that require computations in the configuration space (see, e.g., the work of Avnaim et al. [4]).

Both piano-movers' algorithms are based on the definition of planar *critical curves* and the ability to construct their arrangement, namely the subdivision to maximally-connected component they induce on the planar workspace. In this report we concentrate on the algebraic definition of the critical curves and strive to achieve a compact representation for them. We give a detailed analysis of the critical curves in case of a general polygonal robot, omitted from the publicly available version of the paper [12] (it can only be found in an old technical report by Schwartz and Sharir [11]), and show that under the assumption that the input is given using rational coordinates, we can represent the critical curves as algebraic curves with rational (or, equivalently, integer) coefficients. We also examine the possibility of representing the arrangement of the curves using number types that support exact algebraic computations.

The rest of this report is organized as follows: In Section 2 we discuss the importance of exact computation for the piano-movers' algorithms and mention the available software libraries that can perform exact and robust computation. In Section 3 and 4 we review the piano-movers' algorithm for a ladder and for a general polygon. We thoroughly analyze the critical curves in each case, and for the sake of completeness also bring all other details of the two motion-planning algorithms. Finally, we give some concluding remarks in Section 5.

2 Exact Algebraic Computation

The main focus in the work of Schwartz and Sharir was to give a polynomial algorithm for the fundamental motion-planning problem on the plane. They showed that in both the ladder case and the polygonal robot case the critical curves are algebraic curves of degree 4 at most, therefore the maximal number of intersections between any pair of critical curves is bounded by a constant (16 in this case). As there are $O(n^2)$ critical curves in case of a ladder, where n is the total number of vertices of the polygonal obstacles, and $O(m^2 n^2)$ critical curves in case of a polygonal robot, where m is the number of robot vertices, the complexity of the arrangement of these curves is $O(n^4)$ and $O(m^4 n^4)$, respectively. However, they used the "real RAM" model, assuming that each arithmetic or algebraic operation, such as computing the intersections between two critical curves or comparing the x-coordinates of two such intersection points, can be carried out in constant time.

Bañon[5] presented an implementation for the "ladder" algorithm. However, Bañon's work uses floating-point arithmetic as its underlying number type, making the implementation inexact. Thus, it may fail to find a path if it involves going through tight passages. If the input is not in "general position" and contains degeneracies, or even if it is nearly degenerate, the output may be inconsistent or inaccurate.

In order to give an exact implementation of the two piano-movers' algorithms, one has to be able to construct the planar arrangement of the critical curves in order to capture the true subdivision they induce on the workspace. The arrangement package of CGAL [1, 8], the Computational Geometry Algorithms' Library, can handle the robust construction of such planar arrangements, as long as the users supply a so-called *traits* class that provides all the geometric predicates and constructions for the critical curves, and does so in an exact manner. To this end, one should be able to construct new points — namely to compute the intersections between critical curves and to find a finite set of points that subdivide a critical curve into x-monotone sub-curves — and also be able to evaluate exact predicates involving these points, e.g. to compare the x-coordinates of two points, or to determine whether a given point is above or below a given x-monotone sub-curve.

Schwartz ans Sharir [11] show that all critical curves are *algebraic curves* of degree 4 at most — that is, each critical curve γ is of the form $\gamma(x, y) = \sum_{0 \le i+j \le 4} c_{ij} x^i y^j$ where $c_{ij} \in \mathbb{R}$. In order to find the intersection points between two given algebraic curves $\gamma_1, \gamma_2 \in \mathbb{R}[x, y]$, we wish to calculate the solution for the following equation system:

$$\begin{cases} \gamma_1(x, y) = 0 \\ \gamma_2(x, y) = 0 \end{cases} .$$

We can employ techniques of resultant calculus (see, e.g., Cox et al. [7]) to solve the system:

- Compute the resultant polynomial $\xi(x) = \mathrm{res}_y(\gamma_1, \gamma_2)$.
- Compute the roots of $\xi(x)$, which are the x-coordinates of the intersection points.
- Compute the resultant polynomial $\eta(y) = \mathrm{res}_x(\gamma_1, \gamma_2)$.
- Compute the roots of $\eta(y)$, which are the y-coordinates of the intersection points.
- Pair the x and y-coordinates together, i.e. find all pairs (x_i, y_j) such that $\gamma_1(x_i, y_j) = 0$ and $\gamma_2(x_i, y_j) = 0$.

To subdivide an algebraic curve $\gamma \in \mathbb{R}[x, y]$ into x-monotone sub-curves it is necessary to locate all the points in its interior where the tangent to the curve is a vertical line. In this case, we can use the same techniques we used in the case of intersection computation in order to solve the following system:

$$\begin{cases} \gamma(x,y) = 0 \\ \frac{\partial \gamma}{\partial x}(x,y) = 0 \end{cases} .$$

2.1 Working with Rational Curves

Let us assume that all the critical curves we computed have rational coefficients (that is, they are in $\mathbb{Q}[x, y] \subset \mathbb{R}[x, y]$). As the resultant is defined as the determinant of a square matrix of polynomials with rational coefficients in this case, it yields a univariate polynomial with rational coefficients. We thus have that $\xi, \eta \in \mathbb{Q}[x]$, and we can easily find equivalent polynomials with *integer* coefficients.

The efforts towards providing software packages for robust and efficient computation with algebraic numbers[1] are led by two groups. The LEDA [3] and CORE [2] libraries both offer number types (`leda_real` in LEDA and `EXPR` in CORE) which can be constructed from an integer or a rational number and support all the arithmetic operators ($+, -, \times$ and $\div$) as well as the radical ($\sqrt[k]{\cdot}$) operator. Both libraries use a similar representation for the numbers and rely on the theory of separation bounds to perform floating-point filtering and to guarantee the correctness of the computation — see [6, 10] for more details on the theoretical background underlying the libraries for exact computation. The recent version of the CORE library also offer so-called *root operator* (or a *diamond operator*), where $\diamond(k, a_0, a_1, \ldots, a_n)$ represents the kth-largest real-valued root of the polynomial $p(x) = \sum_{i=0}^{n} a_i x^i$, where $a_0, \ldots, a_n \in \mathbb{Z}$. It is therefore possible to extract a root of any polynomial with integer coefficients and use its value in other calculations, such that the result will still be accurate.

The CORE library with its number-types (and hopefully future versions of the LEDA library) can serve as a convenient platform for the implementation of an arrangement traits-class for critical curves, providing that all our curves really have rational coefficients. In Section 3 and in Section 4 we will show that if the input contains only rational numbers, then we can represent all the critical curves as *algebraic curves* with *rational coefficients* of degree 4 at most. As a consequence, the coordinates of the intersection points and of the vertical tangency points are roots of polynomials with integer coefficients of degree 16 and 12, respectively.[2]

In [13] we give a similar analysis of the curves that support the edges of the Voronoi diagram of a set of polygons with rational coordinates and of the Minkowski sums of these polygons with a disc whose squared radius is rational. We show there that all these curves can be represented as algebraic curves with rational coefficient of degree 2 at most, and use this fact to devise a robust implementation of the arrangement of such curves.

[1] A real number $\alpha \in \mathbb{R}$ is called *algebraic* if and only if there exists a polynomial $p(x) = \sum_{i=0}^{d} a_i x^i$ with integer coefficients such that $p(\alpha) = 0$. If p divides any other polynomial $\tilde{p} \in \mathbb{Z}[x]$ such that $\tilde{p}(\alpha) = 0$, we say that α is an algebraic number of *degree* $d = \deg(p)$.

[2] Note that if we have a polynomial with algebraic irrational coefficients, its roots are also algebraic. However, we need a polynomial with a higher degree to represent these roots as roots of a polynomial with integer coefficients. For example, the root of the linear polynomial $p(x) = \sqrt{2}x - \sqrt[3]{5}$ is a root of the polynomial $\tilde{p}(x) = 8x^6 - 25$, and indeed it is an algebraic number of degree 6.

3 Motion Planning for a Ladder

The simplest form of a robot whose motion in the plane has three degrees of freedom is a line segment pq (also called a "ladder"), which is free to translate and rotate. A *configuration* c of the robot is given by $\langle x, y, \theta \rangle$, where (x, y) is the location of p and θ is the angle between the vector $\vec{pq}$ and the x-axis. A configuration c is called *free* if the ladder does not collide with any of the obstacles when placed at c. If it just touches an obstacle when placed at c, but does not penetrate any obstacle, c is called *semi-free*. If the robot placed at c penetrates some obstacle, the configuration is called *forbidden*.

In addition, we are given a set of polygons with a total number of n vertices, called the *obstacles*. Our goal is to preprocess the input scene, so we can answer motion-planning queries efficiently: Given a start and a goal configuration (both are free configurations of course), we should find collision-free motion path for the ladder, such that the robot does not penetrate any of the obstacles when moving along this path, or determine that no such path exists.

The main idea in the piano-movers algorithm is to subdivide the plane into a finite number of cells, such that each cell can be given a combinatorial label. We use this subdivision to construct a graph that captures the connectivity of the free configuration space. The boundaries of the cells are determined by a set of critical curves, defined by the set of configurations where specific ladder features (a vertex or the interior of the ladder) are in contact with specific obstacle features (a vertex or an edge). In Section 3.1 we give a geometric definition of critical curves, and describe the combinatorial construction of the connectivity graph in Section 3.2.

3.1 Definition of Critical Curves

We will assume that as an input we get a set of simple polygonal obstacles $P_1, \ldots, P_k$, such that each polygon is given as a sequence of points with *rational coordinates*. The length of the ladder, denoted by ℓ, is also given as part of the input and we will also assume that $\ell \in \mathbb{Q}$.

We now define a set of critical curves. First, each obstacle edge defines a critical curve which is obviously a line segment. If an obstacle edge is defined by the endpoints $v_1 = (x_{v_1}, y_{v_1})$ and $v_2 = (x_{v_2}, y_{v_2})$, then its supporting line is given by $(y_{v_1} - y_{v_2})x + (x_{v_2} - x_{v_1})y + (x_{v_1}y_{v_2} - x_{v_2}y_{v_1}) = 0$ and is therefore an algebraic curve with rational coefficients of degree 1. In addition, the following critical curves are defined (see Figure 1):

Type I: For each obstacle edge E, the curve traced by p as q slides along E. This curve is a line segment parallel to E and at distance ℓ from it.

Type II: For each obstacle vertex v, the curve traced by p as q touches v and the ladder rotates around it. This curve is a circular arc centered at v and whose radius is ℓ, bounded by the two lines supporting the two obstacle edges that intersect at v.

Type III: For each obstacle edge E with a convex vertex v as its endpoint, the curve traced by p as q slides from v along E. This curve is a line segment of length ℓ that lies on the same line as E with v being its endpoint.

Type IV: For each pair of convex obstacle vertices v_1 and v_2 that are not two endpoints of a common obstacle edge and such that $d = ||v_1 - v_2|| < \ell$, the curve traced by p as the ladder slides while touching both vertices and initially q touches v_1. This curve is a line segment of length $\ell - d$ that lies on the same line connecting v_1 and v_2 with v_2 as its endpoint.

Type V: For each obstacle edge E and a vertex v that is not one of E's endpoints and whose distance d from E is less than ℓ, the curve tranced by p as q slides along E and the interior of the ladder touches v. This curve is the loop of the *conchoid of Nicomedes*, which is an algebraic curve of degree 4, as we will show in Section 3.1.4.

3.1.1 Critical Curves of Type I

A critical curve of type I is a line segment parallel to an obstacle edge at a distance ℓ from this edge. The supporting line of this curve is therefore parallel to the line supporting the obstacle edge, which obviously has rational coefficients, and lies at a distance ℓ from it — however, this line cannot be represented in general as a linear curve with rational coefficients.[3] However, given a line $ax + by + c = 0$ (with $a, b, c \in \mathbb{Q}$) we can formulate the locus of all points at a distance ℓ from that line:

$$\begin{aligned} \frac{|ax + by + c|}{\sqrt{a^2 + b^2}} &= \ell \\ \frac{(ax + by + c)^2}{a^2 + b^2} &= \ell^2 \end{aligned}$$

$$a^2x^2 + b^2y^2 + 2abxy + 2axc + 2bcy + (c^2 - \ell^2(a^2 + b^2)) = 0 \quad . \tag{1}$$

We obtained an equation of a curve of degree 2 (a *conic curve*) with rational coefficients that represents a pair of parallel lines (actually it is a degenerate parabola). We can therefore represent a critical curve of type I as a curve of degree 2, along with its two endpoints (see also [13] for a similar result). We stress that the endpoints need not have rational coordinates (and in general do not have rational coordinates) — we only need the supporting curve to have rational coefficients.

3.1.2 Critical Curves of Type II

A critical curve of type II is a circular arc of radius ℓ centered at an obstacle vertex $v = (x_v, y_v)$. The supporting curve in this case is obviously:

$$(x - x_v)^2 + (y - y_v)^2 = \ell^2 \quad . \tag{2}$$

Once again, both endpoints of the circular arc are given by the intersection of the supporting circle with two rays emanating from v, and their coordinates are algebraic numbers of degree 2. This does not affect our analysis, as the important observation is that the supporting curve indeed has rational coefficients.

3.1.3 Critical Curves of Type III and IV

A critical curve of types III is a continuation of an obstacle edge and therefore lies on the same supporting line as this edge. A curve of type IV lies on the straight line connecting two obstacle vertices. In both cases, the supporting curve can be represented as a line with rational coefficients.

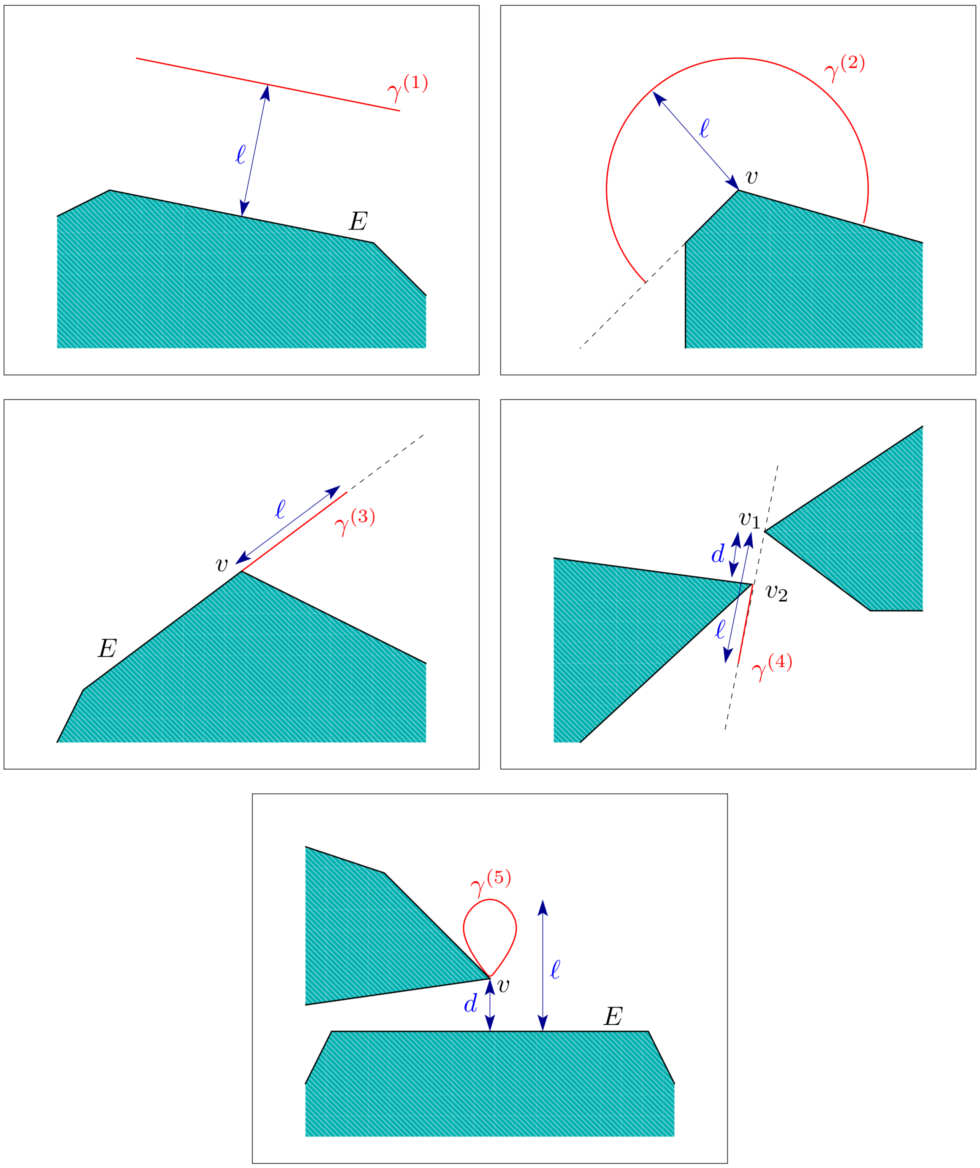

Figure 1: The critical curves for translation and rotation of a ladder in the plane: $\gamma^{(1)}, \ldots, \gamma^{(5)}$ represent critical curves of types I–V, respectively.

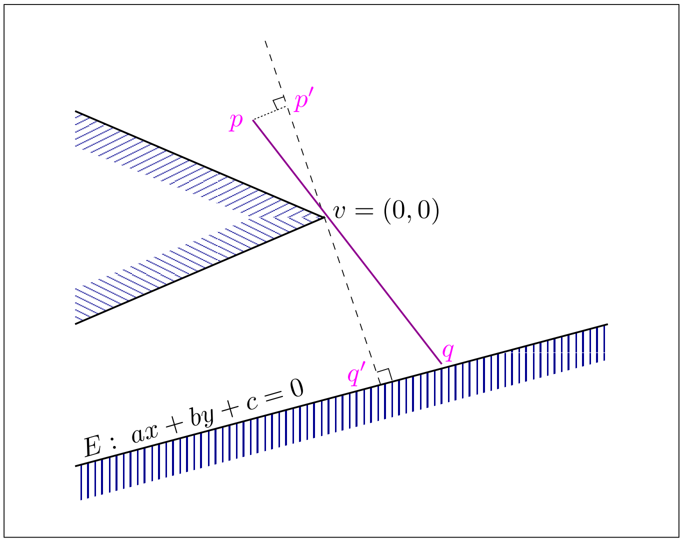

Figure 2: Analysis of a critical curve of type V.

3.1.4 Critical Curves of Type V

Let the ladder vertex p be located at (x, y) such that q touches the obstacle edge E and the interior of the ladder touches the convex obstacle vertex v. Let us assume for simplicity that $v = (0, 0)$. Furthermore, let $ax + by + c = 0$ be the line supporting the edge E. If we denote the x-coordinate of q by σ, we have $q = (\sigma, -\frac{a\sigma+c}{b})$, assuming without loss of genericity that $b \neq 0$ (otherwise we can rotate the scene by 90° and swap the roles of x and y).

If we pass a line perpendicular to E that passes through the origin v, this line obviously has the equation $bx - ay = 0$. Let p' and q' be the projection of p and q onto this line, respectively (see Figure 2 for an illustration). We therefore have:

$$p' = \left(-\frac{a(ax+by)}{a^2+b^2}, -\frac{b(ax+by)}{a^2+b^2}\right) , \qquad q' = \left(-\frac{ac}{a^2+b^2}, -\frac{bc}{a^2+b^2}\right) \quad .$$

The (signed) distances of these two points from the origin are given by:

$$\begin{aligned} \|p'\| &= \sqrt{\frac{a^2(ax+by)^2}{(a^2+b^2)^2} + \frac{b^2(ax+by)^2}{(a^2+b^2)^2}} = \frac{ax+by}{\sqrt{a^2+b^2}} \quad , \\ \|q'\| &= \sqrt{\frac{a^2c^2}{(a^2+b^2)^2} + \frac{b^2c^2}{(a^2+b^2)^2}} = \frac{c}{\sqrt{a^2+b^2}} \quad . \end{aligned}$$

We can also compute the (signed) distances of p and p and the line $bx - ay = 0$:

$$\begin{aligned} \|p - p'\| &= \frac{bx - ay}{\sqrt{a^2+b^2}} \quad , \\ \|q - q'\| &= \frac{b\sigma + a\frac{a\sigma+c}{b}}{\sqrt{a^2+b^2}} = \frac{(a^2+b^2)\sigma + ac}{b\sqrt{a^2+b^2}} \quad . \end{aligned}$$

[3]Consider, for example, the line parallel to $y = x$ and at distance 1 from it — there are actually two such lines, $y = x \pm \sqrt{2}$, neither of them can be represented using rational coefficients.

Now since the triangles $\triangle vpp'$ and $\triangle vqq'$ are similar, and since x and σ have opposite signs, we have:

$$\frac{\|p - p'\|}{\|p'\|} = -\frac{\|q - q'\|}{\|q'\|} \quad . \tag{3}$$

Substituting the expressions for these distances and reducing the $\sqrt{a^2 + b^2}$ factor, we obtain:

$$\begin{aligned} \frac{bx - ay}{ax + by} &= -\frac{(a^2 + b^2)\sigma + ac}{bc} \\ (ax + by)(a^2 + b^2)\sigma &= (a^2 + b^2)cx \\ \sigma &= \frac{cx}{ax + by} \quad . \end{aligned} \tag{4}$$

Having expressed σ in terms of x and y we can write the equation expressing the length of the ladder (namely, $\|p - q\|^2 = \ell^2$):

$$\begin{aligned} (x - \sigma)^2 + \left(y + \frac{a\sigma + c}{b}\right) &= \ell^2 \\ b^2\left(x - \frac{cx}{ax + by}\right)^2 + \left(by + \frac{cx}{ax + by} + c\right)^2 &= b^2\ell^2 \\ b^2(ax^2 + bxy - cx)^2 + (abxy + b^2y^2 + 2acx + bcy)^2 &= b^2\ell^2(ax + by)^2 \quad . \end{aligned} \tag{5}$$

We found that the locus where p can be located such that q touches E and the ladder touches v is an algebraic curve of degree 4 (a *quartic curve*). In case v is not located at the origin, but at (x_v, y_v), one can simply substitute each x by $(x - x_v)$ and $(y - y_v)$ into the equation above, so that the degree of the curve remains unchanged. Finally, since $x_v, y_v, a, b, c, \ell \in \mathbb{Q}$, its is clear that our quartic curve has rational coefficients.

Our analysis generalizes the analysis given in [9, 12], where E lies on the horizontal line $y = -d$, where d is the distance between v and E.

Observation 1 *It is important to notice that in all cases where the* squared *length of the ladder ℓ takes part in the definition of a supporting curve (see equations (1), (2) and (5) above). It is therefore possible to relax the requirement that $\ell \in \mathbb{Q}$ and instead assume that only ℓ^2 is rational.*

3.2 The Combinatorial Construction

The next step in the piano-movers' algorithm is to construct the arrangement of all critical curves, as defined in the previous section. Having done so, we can pick some point at the interior of a certain arrangement cell C. If we locate the vertex p at this point, we still have one degree of freedom for the ladder, namely its orientation, or the angle θ it forms with the x-axis. If we let this angle vary from $0°$ to $360°$, we shall come across a finite number of so-called *critical angles* where either of the following happens:

- The vertex q touches an obstacle edge.
- An obstacle vertex touches the segment pq.

Thus, we can label each critical angle with the obstacle feature (either a vertex or an edge) associated with it. Furthermore, each pair of critical angles define a free sub-domain of free orientations for the robot at the chosen position.

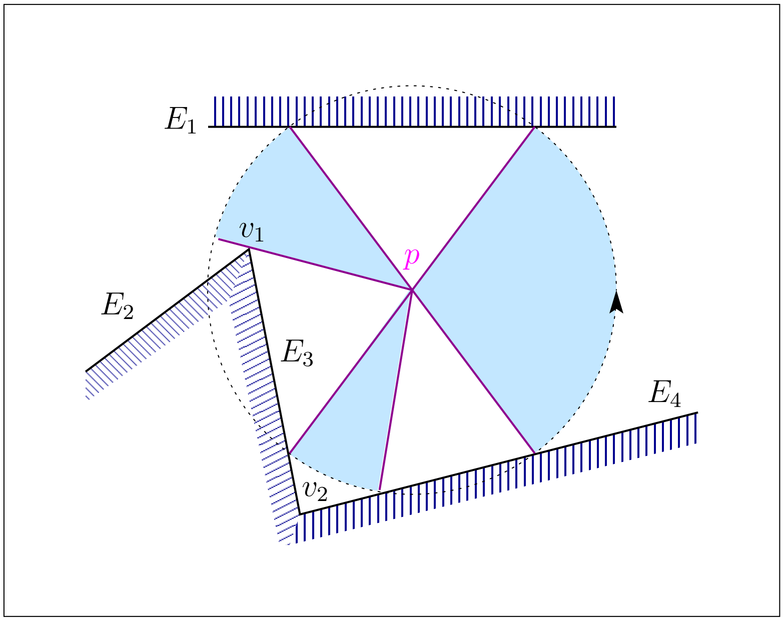

Figure 3: The legal orientation ranges for a given placement of the ladder vertex p (shaded). Notice the obstacle features (vertices and edges) that define critical angles.

Now if we pick another position for the ladder within the same cell C, the critical angles will change, but the definition of the critical curves ensures that their labeling will remain the same. Thus, for each cell C is the arrangement, we can define the *characteristic label* of the cell, denoted $\chi(C)$. In case that all the orientations within a certain cell are illegal, we have of course $\chi(C) = \emptyset$. On the other hand, if all the orientations within a cell are free, we use the notation $\chi(C) = \big\{[\Omega, \Omega]\big\}$, where Ω stands for a non-existing feature.

For example, in Figure 3 the characteristic label of the cell that the current position of p belongs to is $\big\{[E_1, v_1], [E_3, E_4], [E_4, E_1]\big\}$.

Given a position $(x, y) \in C$ and $[f_1, f_2] \in \chi(C)$, let us denote the two critical angles related to the two obstacle features f_1, f_2 by $\lambda_l^{(C)}(x, y, f_1)$ and $\lambda_h^{(C)}(x, y, f_2)$. We shall also use the convention $\lambda_l(x, y, \Omega) = 0°$ and $\lambda_h(x, y, \Omega) = 360°$ for any cell. We can now define a three-dimensional free cell κ of as:

$$\kappa = \mathrm{cell}(C, f_1, f_2) = \left\{ \ \langle x, y, \theta \rangle \ \ | \ \ (x, y) \in C, \ \ \theta \in \big(\lambda_l^{(C)}(x, y, f_1), \lambda_h^{(C)}(x, y, f_2)\big) \ \right\} \quad .$$

We shall now construct a graph $\mathcal{G}$ with the three-dimensional cells of free configurations as its vertices. In this graph, we shall connect two cells $\kappa = \mathrm{cell}(C, f_1, f_2)$ and $\kappa' = \mathrm{cell}(C', f_1', f_2')$ if and only if the two-dimensional cells C and C' share a segment γ of a critical curve at their boundary and:

$$\forall (x, y) \in \gamma \quad \big(\lambda_l^{(C)}(x, y, f_1), \lambda_h^{(C)}(x, y, f_2)\big) \cap \big(\lambda_l^{(C')}(x, y, f_1'), \lambda_h^{(C')}(x, y, f_2')\big) \neq \emptyset \quad .$$

In fact, for each neighboring two-dimensional cells C and C' we shall do the following:

- For each $[f_1, f_2] \in \chi(C) \cap \chi(C')$ we shall connect $\mathrm{cell}(C, f_1, f_2)$ and $\mathrm{cell}(C', f_1, f_2)$.
- For each $[f_1, f_2] \in \chi(C) \setminus \chi(C')$ we shall connect $\mathrm{cell}(C, f_1, f_2)$ with all other cells $\kappa' = \mathrm{cell}(C', f_1', f_2')$ that are induced by C'.

- For each $[f_1', f_2'] \in \chi(C') \setminus \chi(C)$ we shall connect $\text{cell}(C', f_1', f_2')$ with all other cells $\kappa = \text{cell}(C, f_1, f_2)$ that are induced by C.

We are now ready to answer motion-planning queries using the connectivity graph $\mathcal{G}$. Given a start configuration c_s and a goal configuration c_t, we locate the two cells κ_s and κ_t that contain these two configurations and check whether they are connected in $\mathcal{G}$. If not, we have a combinatorial disconnection proof. Otherwise, we can trace one of the paths that connect the two cells are devise a collision-free motion for the ladder.

4 Motion Planning for a General Polygon

The case of a general polygon which is free to translate and rotate amidst polygonal obstacles is more complicated. However, the same notions and techniques developed for the case of a ladder can be extended to deal with the general case.

As in the ladder case, we will assume that the obstacles are simple polygons, such that all obstacle vertices have rational coordinates. In addition, we are given a polygonal robot with m vertices, all of them having rational coordinates. One of the vertices will be denoted p and serve as our reference point, while the other robot vertices will be denoted $q_1, \ldots, q_{m-1}$. A configuration $c = \langle x, y, \theta \rangle$ of the polygonal robot defines its placement uniquely, where the reference vertex p is located at (x, y) and θ is the angle the vector $\vec{pq_1}$ forms with the x-axis.

The robot has m edges, but in order to generalize the concept of a characteristic label for a cell, as developed in the previous section, we will have to subdivide some of these edges to two *half-edges*. More precisely, for each edge $q_i q_{i+1}$ we shall draw a line perpendicular to the edge that passes through p. If this line intersects the interior of the segment $q_i q_{i+1}$ (that is, if both angles $\angle p q_i q_{i+1}$ and $\angle p q_{i+1} q_i$ are acute) we will use this intersection point to subdivide the edge into two half-edges. We label the half-edges $L_1, L_2, \ldots, L_{m'}$ starting from p in a counterclockwise direction (m' is obviously bounded by $2(m-1)$, so the asymptotic complexity of the robot remains unchanged) — see Figure 4 for an illustration.

It is important to notice that even though we create new obstacle vertices, these vertices still have rational coordinates. Let us assume that we subdivide an obstacle edge L supported by the line $ax + by + c = 0$. We split this edge at the intersection point of this line with a line perpendicular to it that passes through $p = (x_p, y_p)$. As this perpendicular line is given by $bx - ay + (ay_p - bx_p) = 0$, the new vertex v' is the intersection of these two lines (recall that if L's endpoints are q_i and q_j then $a = y_{q_i} - y_{q_j}$, $b = x_{q_j} - x_{q_i}$ and $c = x_{q_i} y_{q_j} - x_{q_j} y_{q_i}$):

$$v' = \left(\frac{b(bx_p - ay_p) - ac}{a^2 + b^2} \, , \, \frac{a(ay_p - bx_p) - bc}{a^2 + b^2} \right) \quad ,$$

and as a, b, c, x_p and y_p are all rational, v' is a point with rational coefficients.

In the next section we shall define the critical curves in the case of a polygonal robot and analyze their algebraic properties.

4.1 Definition of Critical Curves

As we did in the ladder case (see Section 3.1), we first define a set of critical curves based on the structure of the obstacles and the polygonal robot — yet when dealing with a polygonal robot the

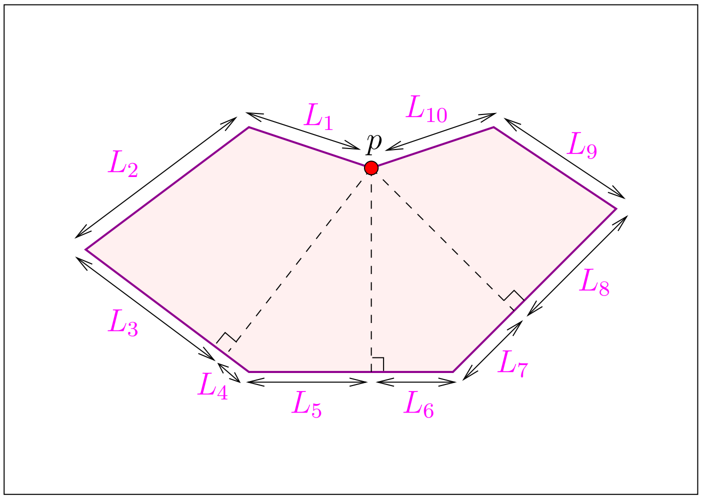

Figure 4: The subdivision of a polygonal robot with 7 edges into 10 half-edges with respect to the reference vertex p.

degenerate semi-free configurations that induce the critical curves are somewhat more complex. First, each obstacle edge defines a critical curve which is obviously a segment of a straight line with rational coefficients. In addition, the following critical curves are defined:

Type I: For each obstacle edge E and for each robot vertex $q \neq p$, the curve traced by p as q slides along E. This curve is a line segment parallel to E and at distance $||q-p||$ from it.

Type II: For each obstacle vertex v and for each robot vertex $q \neq p$, the curve traced by p as the robot rotates around v and q touches it. This curve is a circular arc whose center is v and radius is $||q-p||$.

Type III: For each obstacle edge E and two convex robot vertices $q_1, q_2 \neq p$ such that $||q_1 - q_2|| < ||E||$, the curve traced by p as q_1, q_2 slide along E. This curve is a line segment of length $||E|| - ||q_1 - q_2||$ that lies parallel to E.

Type IV: For each pair of convex obstacle edges E_1, E_2 and two convex robot vertices $q_1, q_2 \neq p$, the curve traced by p as q_1 slides along E_1 while q_2 slides along E_2. We show in Section 4.1.1 that this curve is an elliptic arc.

Type V: For each obstacle edge E, non-incident convex obstacle vertex v, convex robot vertex $q \neq p$ and robot half-edge L, the curve tranced by p as q slides along E while L touches v. In Section 4.1.2 we show that this defines a segment of an algebraic curve of degree 4.

Type VI: For each two convex obstacle vertices v_1, v_2 and each robot half-edge L that does not contain p, and such that $||v_1 - v_2|| < ||L||$, the curve traced by p as L slides while touching both v_1 and v_2. This curve is a line segment of length $||L|| - ||v_1 - v_2||$ that lies parallel to the line connecting v_1 and v_2.

Type VII: For each two convex obstacle vertices v_1, v_2 and two robot half-edges L_1, L_2, the curve traced by p as L_1 touches v_1 and L_2 touches v_2. We prove in Section 4.1.3 that this defines a segment of an algebraic curve of degree 4.

Type VIII: For each obstacle edge E and each robot half-edge L that does not contain p, the curve traced by p as L slides along E. This curve is a line segment that lies parallel to E.

The supporting lines of critical curves of types I, III, VI and VIII can all be characterized as the locus of all points lying at some distance δ from a given line:

- A critical curve of type I is parallel to an obstacle edge E at a distance of $\delta = \|pq\|$.
- A critical curve of type III is parallel to an obstacle edge E, with δ being the distance between p and the line connecting the robot vertices q_1 and q_2.
- A critical curve of type VI is parallel to the line connecting the obstacle vertices v_1 and v_2, where δ is the distance of p from the robot half-edge L.
- A critical curve of type VII is parallel to the obstacle edge E, lying at a distance δ that equals the distance of p from the robot half-edge L.

In all cases, the base line defining the line supporting the critical curve is either an obstacle edge or a line connecting two obstacle vertices and therefore has rational coefficients. Moreover, the required *squared* distance is also a rational number, as it is either the distance between the reference vertex p and another obstacle vertex, or the distance between p and a line with rational coefficients. We therefore can use the same formulation we used in Section 3.1.1, and obtain a supporting line-pair with rational coefficients — a degenerate conic curve of the form of Equation (1).

The analysis of a type II curve is very similar to the ladder case (see Equation (2)) — we just have to replace ℓ^2 by $\|pq\|^2$, which is obviously a rational number.

We shall next analyze the algebraic representation of the critical curves of types IV, V and VII and parametrize them in terms of the robot features and the location of the obstacles.

4.1.1 Critical Curves of Type IV

We are interested in the location (x, y) of the reference robot vertex p such that the convex robot vertex q_1 touches the obstacle edge E_1 and the convex robot vertex q_2 touches the obstacle edge E_2.

For our convenience, let us define a coordinate system $\mathcal{C}_R$ associated with the robot, such that q_1 is its origin and q_2 lies on it x-axis. If the distance between these two robot vertices is d, then the location of q_2 under $\mathcal{C}_R$ is $(0, d)$. Let us denote the location of the reference vertex p under this coordinate system by (x_0, y_0). To avoid confusion, we shall refer to these locations as q'_1, q'_2 and p', respectively.

We shall also associate a second coordinate system $\mathcal{C}_S$ with the obstacle scene, such that E_1 coincides with its y-axis (that is, it lies on the line $x = 0$) and E_2 lies on a line passing through the origin (that is, it lies on the line $y = ax$ where a is a constant that depends only on the shapes and placements of the obstacles in the scene). See Figure 6 for an illustration.

Now let us define a rigid transformation $T : \mathcal{C}_R \to \mathcal{C}_S$ in the following manner: Pick a point on E_1, whose coordinates under $\mathcal{C}_S$ are given by $(0, \sigma)$ and move the robot so q'_1 coincides with it. Now rotate the robot around this point by φ until q'_2 becomes in contact with E_2. We shall denote the coordinates of $q_2 = T(q'_2)$ by $(\tau, a\tau)$. Let (x, y) be the location of p under $\mathcal{C}_S$ after this transformation.

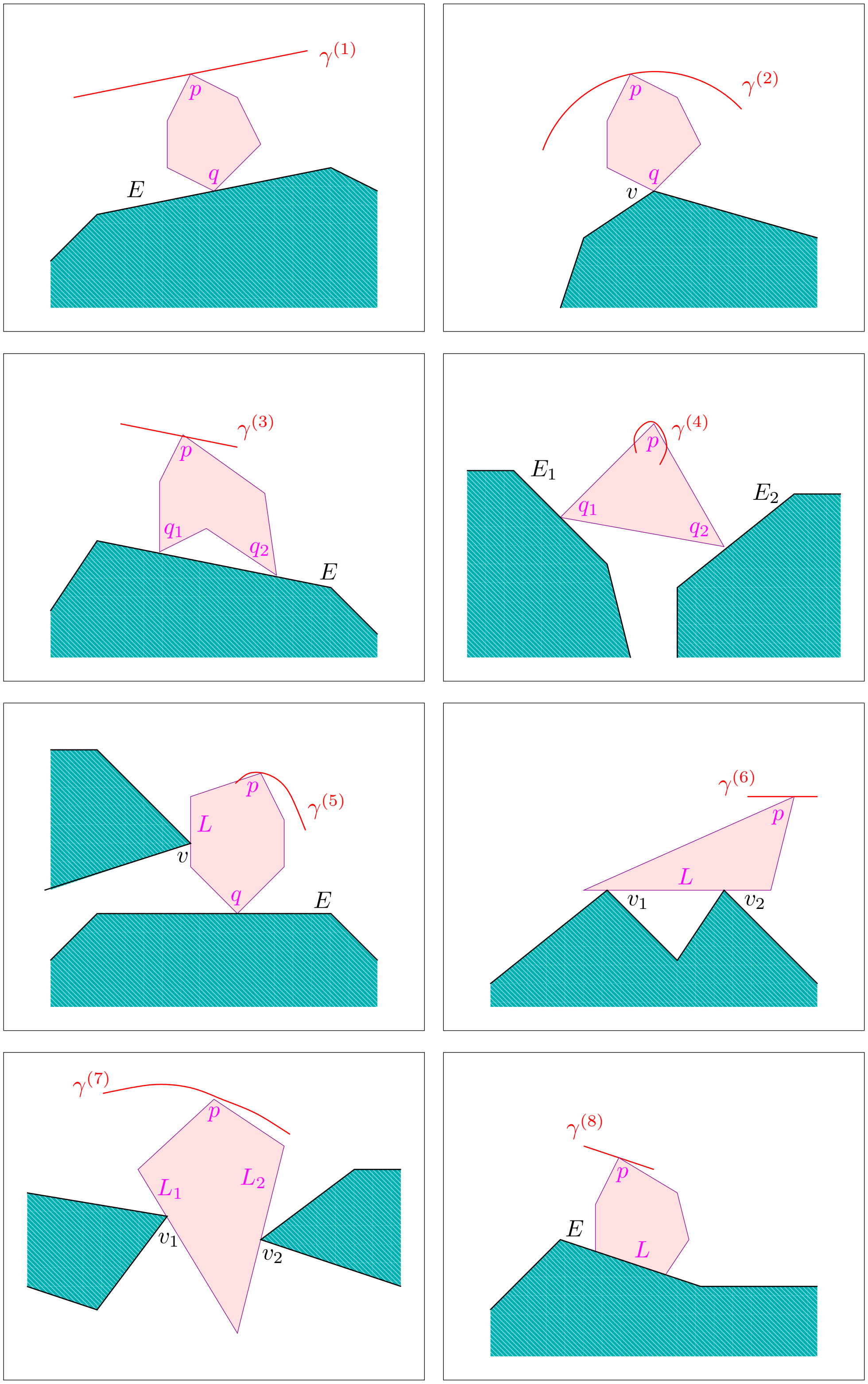

Figure 5: The critical curves for translation and rotation of a polygon in the plane: $\gamma^{(1)}, \ldots, \gamma^{(8)}$ represent critical curves of types I–VIII, respectively.

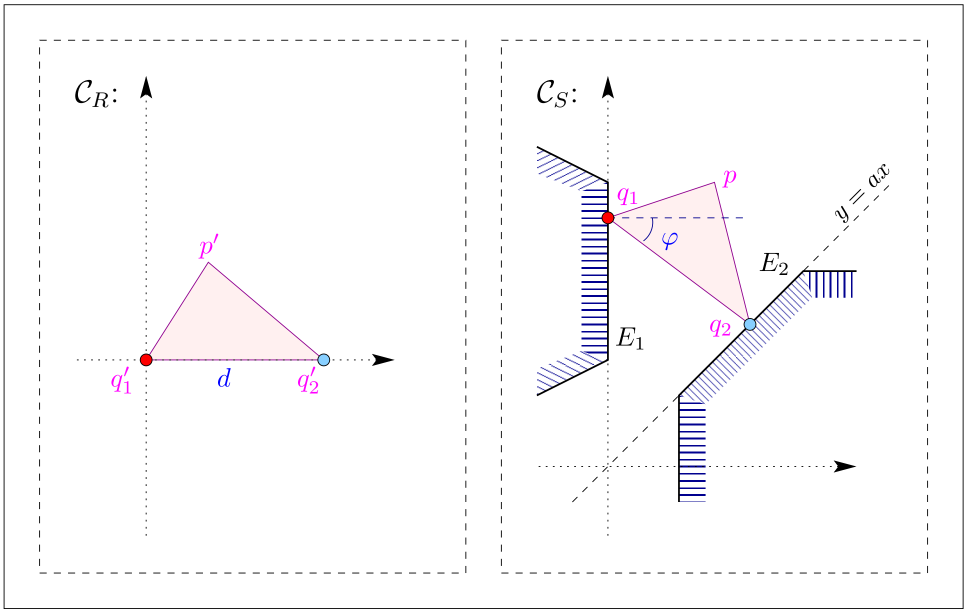

Figure 6: Analysis of a critical curve of type IV.

It is clear that since T preserves distances, then $||q_2 - q_1|| = ||q'_2 - q'_1||$, so we have:

$$\tau^2 + (a\tau - \sigma)^2 = d^2 \quad . \tag{6}$$

Furthermore, since the robot is not deformed by the rigid transformation, all angles are preserved and in particular $\angle p'q'_1q'_2 = \angle pq_1q_2$. Hence, the vector $\vec{q'_1p'}$ equals the vector $\vec{q_1p}$ rotated by $-\varphi$ (where φ is the angle between the vector $\vec{q_1q_2}$ and $\mathcal{C}_S$'s x-axis). Thus:

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} x \\ y - \sigma \end{pmatrix} \quad . \tag{7}$$

But it is clear that (see Figure 6):

$$\cos\varphi = \frac{\tau}{\sqrt{\tau^2 + (a\tau - \sigma)^2}} = \frac{\tau}{d} \quad , \qquad \sin\varphi = \frac{a\tau - \sigma}{\sqrt{\tau^2 + (a\tau - \sigma)^2}} = \frac{a\tau - \sigma}{d} \quad .$$

Hence (7) can be written as (note that we have inverted the rotation matrix):

$$\frac{1}{d} \begin{pmatrix} \tau & \sigma - a\tau \\ a\tau - \sigma & \tau \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} x \\ y - \sigma \end{pmatrix} \quad . \tag{8}$$

We therefore obtain a linear system of equations in σ and τ:

$$\begin{cases} y_0\sigma + (x_0 - ay_0)\tau = dx \\ (d - x_0)\sigma + (ax_0 + y_0)\tau = dy \end{cases} \quad . \tag{9}$$

For now we shall assume that the rank of this system is 2 (in the next subsection we focus on the degenerate case when its rank is 1), namely the matrix of the system coefficients is not singular as $\left(x_0^2 + y_0^2 + d(ay_0 - x_0)\right) \neq 0$. To solve our linear system, we can simply invert the matrix of coefficients, and compute σ and τ:

$$\begin{aligned} \begin{pmatrix} \sigma \\ \tau \end{pmatrix} &= \frac{1}{x_0^2 + y_0^2 + d(ay_0 - x_0)} \begin{pmatrix} ax_0 + y_0 & ay_0 - x_0 \\ x_0 - d & y_0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \\ &= \frac{d}{x_0^2 + y_0^2 + d(ay_0 - x_0)} \begin{pmatrix} a(x_0x + y_0y) + (y_0x - x_0y) \\ (x_0x + y_0y) - dx \end{pmatrix} \quad . \end{aligned} \tag{10}$$

We can now substitute the expressions for σ and τ into Equation (6), and obtain (after multiplying by the denominator):

$$\begin{aligned} & d^2\left((x_0x + y_0y) - dx\right)^2 + \\ + \; & d^2\left(a(x_0x + y_0y) - adx - a(x_0x + y_0y) - (y_0x - x_0y)\right)^2 = \\ & \qquad = d^2\left(x_0^2 + y_0^2 + d(ay_0 - x_0)\right)^2 \quad . \end{aligned}$$

This expression can be simplified to yield the following equation of a conic curve — that is, an algebraic curve of degree 2:

$$\left((x_0x + y_0y) - dx\right)^2 + \left(adx + (y_0x - x_0y)\right)^2 = \left(x_0^2 + y_0^2 + d(ay_0 - x_0)\right)^2$$

$$\left((x_0 - d)^2 + (y_0 + ad)^2\right)x^2 + (x_0^2 - y_0^2)y^2 - 2d(ax_0 + y_0)xy = \left(x_0^2 + y_0^2 + d(ay_0 - x_0)\right)^2 \quad . \tag{11}$$

It is possible to classify the conic curve defined by Equation (11) by looking at the sign of the following expression, involving the coefficients of x^2, y^2 and xy:

$$\begin{aligned} & 4\left((x_0 - d)^2 + (y_0 + ad)^2\right)(x_0^2 - y_0^2) - \left(2d(ax_0 + y_0)\right)^2 = \\ & \qquad = 4\left(x_0^2 + y_0^2 + d(ay_0 - x_0)\right)^2 \quad . \end{aligned}$$

Since $\left(x_0^2 + y_0^2 + d(ay_0 - x_0)\right) \neq 0$ (recall this is the determinant of the matrix of coefficients of the linear system (9)), this expression is always positive — thus we can conclude that the curve is an ellipse.

The Degenerate Case

We have already mentioned that in case that $x_0^2 + y_0^2 = d(ay_0 - x_0)$, the rank of the system (9) is 1, so we cannot proceed and extract the expressions for σ and τ as we did in the general case.

First, let us examine when does such a degenerate case occur. Since we have $x_0^2 + y_0^2 = d(ay_0 - x_0)$, then obviously:

$$a = -\frac{x_0(x_0 - d) + y_0^2}{dy_0} \quad . \tag{12}$$

Since E_1 forms a right angle with the x-axis of $\mathcal{C}_S$ and a is the slope of E_2's supporting line, it equals $\tan(90° - \psi)$, where ψ is the angle between E_1 and E_2 (see Figure 7). We therefore have:

$$\tan\psi = \cot(90° - \psi) = \frac{1}{a} = -\frac{dy_0}{x_0(x_0 - d) + y_0^2} \quad . \tag{13}$$

Now let us examine the triangle $\triangle q_1q_2p$, and let us denote by α_1, α_2 and β the triangle angles at q_1, q_2 and p respectively. By examining the triangle under $\mathcal{C}_R$ it is easy to see that (see Figure 7):

$$\tan\alpha_1 = \frac{y_0}{x_0} \quad , \qquad \tan\alpha_1 = \frac{y_0}{d - x_0} \quad .$$

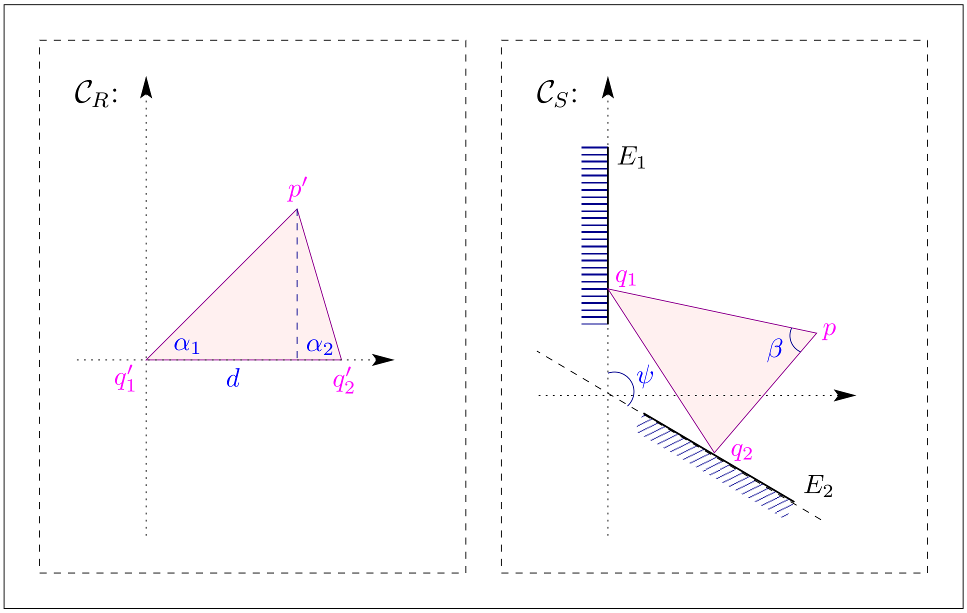

Figure 7: Analysis of the degenerate case for a critical curve of type IV.

And so:

$$\tan\beta = \tan\big((90^\circ - \alpha_1) + (90^\circ - \alpha_2)\big) = \frac{\cot\alpha_1 + \cot\alpha_2}{1 - \cot\alpha_1 \cot\alpha_2} = \frac{dy_0}{x_0(x_0 - d) + y_0^2} \quad . \tag{14}$$

Since $\tan\psi = -\tan\beta$ and both angles must be less than 180°, we can conclude that the degenerate case occurs when $\psi = 180^\circ - \beta$. In this case, the critical curve lies on the line cutting ψ into two angles that equal $\alpha 1$ and α_2. Since the angle between the vector connecting $\mathcal{C}_S$'s origin and p and the y-axis must equal α_2, we have:

$$\tan\alpha_2 = \frac{x}{y} = \frac{y_0}{d - x_0} \quad , \tag{15}$$

so the critical curve in this degenerate case is a segment of the following line:

$$(d - x_0)x - y_0 y = 0 \quad . \tag{16}$$

However, in the next subsection we show that it is more convenient to represent the supporting curve as a conic curve also in the degenerate case. To this end, we multiply Equation (16) by the equation of a perpendicular line passing through $\mathcal{C}_S$'s origin, which is given by $y_0 x + (d - x_0)y = 0$. We therefore obtain the line-pair:

$$(d - x_0)y_0(x^2 - y^2) + \big((d - x_0)^2 - y_0^2\big)xy = 0 \quad . \tag{17}$$

Further Analysis

Up to this point, we have assumed that the robot and the obstacles are positioned in some canonical setting, in order to make our analysis more elegant and to find the algebraic degree of the critical curve. To obtain the true equation of the underlying ellipse under the original coordinate system of the workspace we still have to apply some rigid transformation on the resulting curves. We next show that these rigid transformations provide us with the desired rational coefficients, while they obviously do not effect the algebraic degree of the curve.

So far we obtained the representation in Equation (11), where d is the distance between q_1 and q_2, (x_0, y_0) are the coordinates of p under the coordinate system $\mathcal{C}_R$ and a is the slope of the obstacle edge E_2 under the coordinate system $\mathcal{C}_S$.

Let us assume that the relevant robot vertices are given with the input coordinates (x_p, y_p), (x_{q_1}, y_{q_1}) and (x_{q_2}, y_{q_2}). We first note that $d = \sqrt{(x_{q_2} - x_{q_1})^2 + (y_{q_2} - y_{q_1})^2}$, so d^2 is obviously rational. To bring the vector $\vec{q_1 q_2}$ to coincide with the x-axis, we have to translate the coordinates by $(-x_{q_1}, -y_{q_1})$ and to rotate it by $-\theta$, where θ is the angle between $\vec{q_1 q_2}$ and the x-axis, using the rotation matrix $R_{-\theta}$. We thus have:

$$R_{-\theta} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} ,$$

where:

$$\cos\theta = \frac{x_{q_2} - x_{q_1}}{d} , \qquad \sin\theta = \frac{y_{q_2} - y_{q_1}}{d} .$$

The coordinates of $p' = (x_0, y_0)$ can be therefore expressed as:

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = R_{-\theta} \begin{pmatrix} x_p - x_{q_1} \\ y_p - y_{q_1} \end{pmatrix} = \frac{1}{d} \begin{pmatrix} (x_{q_2} - x_{q_1})(x_p - x_{q_1}) + (y_{q_2} - y_{q_1})(y_p - y_{q_1}) \\ (y_{q_1} - y_{q_2})(x_p - x_{q_1}) + (x_{q_2} - x_{q_1})(y_p - y_{q_1}) \end{pmatrix} .$$

It is easy to show that if we substitute x_0 and y_0 into Equation (11) (or in Equation (17) in the degenerate case), we get an equation involving just d^2 and the input coordinates (x_p, y_p), (x_{q_1}, y_{q_1}) and (x_{q_2}, y_{q_2}).

We still have to express the slope a. Let us assume that the obstacle edges E_1 and E_2 are defined by the endpoints u_1, v_1 and u_2, v_2, respectively. The angles ω_1 and ω_2 these two edges form with the x-axis in the original coordinate system are therefore given by:

$$\cos\omega_1 = \frac{x_{v_1} - x_{u_1}}{\|E_1\|} , \qquad \sin\omega_1 = \frac{y_{v_1} - y_{u_1}}{\|E_1\|} ,$$
$$\cos\omega_2 = \frac{x_{v_2} - x_{u_2}}{\|E_2\|} , \qquad \sin\omega_2 = \frac{y_{v_2} - y_{u_2}}{\|E_2\|} .$$

The sine and cosine of the angle ω between the two edges are therefore:

$$\cos\omega = \cos(\omega_2 - \omega_1) = \frac{(x_{v_2} - x_{u_2})(x_{v_1} - x_{u_1}) + (y_{v_2} - y_{u_2})(y_{v_1} - y_{u_1})}{\|E_1\| \cdot \|E_2\|} ,$$

$$\sin\omega = \sin(\omega_2 - \omega_1) = \frac{(y_{v_2} - y_{u_2})(x_{v_1} - x_{u_1}) - (x_{v_2} - x_{u_2})(y_{v_1} - y_{u_1})}{\|E_1\| \cdot \|E_2\|} .$$

As E_1 coincides with the y-axis in $\mathcal{C}_R$, we have that a is also a rational number, as we can express it as:

$$a = \tan(90^\circ - \omega) = \cot\omega = \frac{(x_{v_2} - x_{u_2})(x_{v_1} - x_{u_1}) + (y_{v_2} - y_{u_2})(y_{v_1} - y_{u_1})}{(y_{v_2} - y_{u_2})(x_{v_1} - x_{u_1}) - (x_{v_2} - x_{u_2})(y_{v_1} - y_{u_1})} .$$

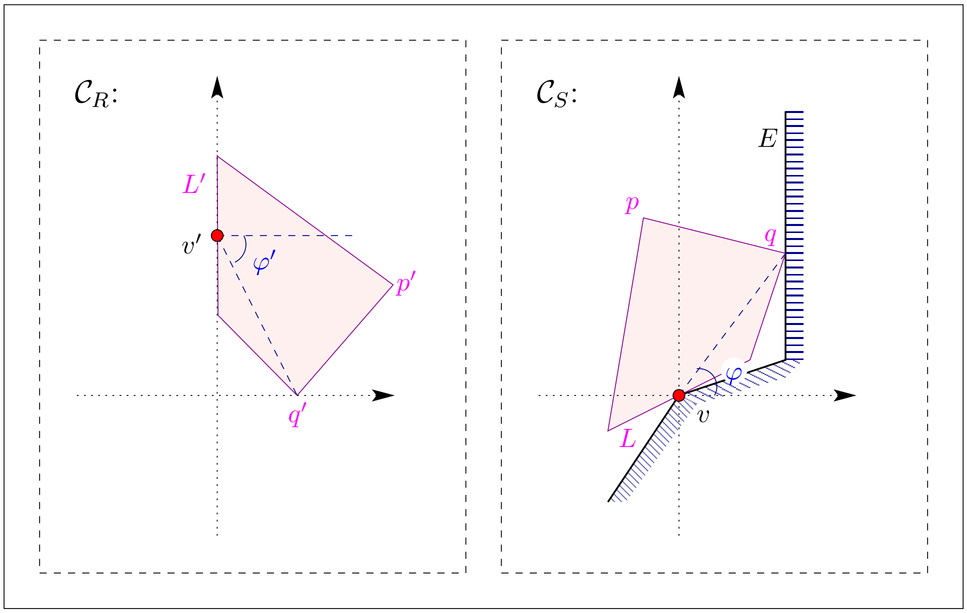

Figure 8: Analysis of a critical curve of type V.

Up to this point, we showed that our ellipse has rational coefficients under the coordinate system $\mathcal{C}_S$. We still have to transform it back to the original coordinate system. We begin by rotating by an angle $\omega_1 - 90°$, or to replace x by $\hat{x}$ and y by $\hat{y}$, where:

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = R_{\omega_1 - 90°} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{\|E_1\|} \begin{pmatrix} (y_{v_1} - y_{u_1})x + (x_{v_1} - x_{u_1})y \\ (y_{v_1} - y_{u_1})y - (x_{v_1} - x_{u_1})x \end{pmatrix} \quad .$$

The length $\|E_1\|$ is irrational, but as all monomials in Equation (11) have an even degree (namely, we have the monomials containing x^2, y^2, xy and a free coefficient), the equation we get only involves $\|E_1\|^2$, which is a rational number. We finally translate the origin of $\mathcal{C}_R$ to the intersection point (x_t, y_t) between the lines supporting E_1 and E_2. This point obviously has rational coordinates, so the translation preserves the "rationality" of the ellipse coefficients.

4.1.2 Critical Curves of Type V

We are interested in the location (x, y) of the reference robot vertex p such that the convex robot vertex q touches the obstacle edge E and the robot half-edge L touches the convex obstacle vertex v (where v is not an endpoint of E).

Let us first assume that q is not an endpoint of the L. We can then associate a coordinate system $\mathcal{C}_R$ with the robot, such that L coincides with its y-axis (that is, it lies on the line $x = 0$) and such that q lies on the x-axis and its coordinates are $(0, d)$, where d is the distance between q and the line containing L. Let (x_0, y_0) denote the coordinates of the vertex p under $\mathcal{C}_R$. For clarity, we shall denote these robot features L', q' and p' respectively under this coordinate system.

We also associate another coordinate system $\mathcal{C}_S$ with the obstacle, with v as its origin and with E being parallel to the y-axis, lying on a line whose equation is $x = \delta$, where δ is the distance between v and the line supporting E (see Figure 8).

Now let us define a rigid transformation $T : \mathcal{C}_R \to \mathcal{C}_S$ in the following manner: Pick a point v' on L', whose coordinates under $\mathcal{C}_R$ are $(0, \sigma)$ and shift the robot such that it coincides with v. Then rotate the robot until q touches the obstacle edge E — at this point, its coordinates under $\mathcal{C}_S$ are (δ, τ). Let (x, y) be the location of p under $\mathcal{C}_S$ after this transformation.

The first observation one can make is that since the rigid transformation preserves distances, we have $||p - v|| = ||p' - v'||$ and $||q - v|| = ||q' - v'||$, hence we can obtain:

$$x^2 + y^2 = x_0^2 + (y_0 - \sigma)^2 \quad , \tag{18}$$

and:

$$\delta^2 + \tau^2 = d^2 + \sigma^2 \quad . \tag{19}$$

The second important observation is that $\angle p'v'q' = \angle pvq$, since the transformation T does not deform the robot and the angle is preserved. Hence, if we rotate the vector $\vec{v'p'}$ by $-\varphi'$ (where φ' is the angle between the vector $\vec{v'q'}$ and $\mathcal{C}_R$'s x-axis) we will get an identical vector to the one obtained by the rotation of $\vec{vp}$ by $-\varphi$ (where φ is the angle between the vector $\vec{vq}$ and $\mathcal{C}_S$'s x-axis). That is, we can write:

$$\begin{pmatrix} \cos\varphi' & \sin\varphi' \\ -\sin\varphi' & \cos\varphi' \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 - \sigma \end{pmatrix} = \begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad . \tag{20}$$

The sines and cosines of φ and φ' are given by (see Figure 8):

$$\cos\varphi' = \frac{d}{\sqrt{d^2 + \sigma^2}} \quad , \qquad \sin\varphi' = \frac{-\sigma}{\sqrt{d^2 + \sigma^2}} \quad ,$$

$$\cos\varphi = \frac{\delta}{\sqrt{\delta^2 + \tau^2}} \quad , \qquad \sin\varphi = \frac{\tau}{\sqrt{\delta^2 + \tau^2}} \quad ,$$

so Equation (20) can be rewritten as:

$$\frac{1}{\sqrt{d^2 + \sigma^2}} \begin{pmatrix} d & -\sigma \\ \sigma & d \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 - \sigma \end{pmatrix} = \frac{1}{\sqrt{\delta^2 + \tau^2}} \begin{pmatrix} \delta & \tau \\ -\tau & \delta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad . \tag{21}$$

Using the equality (19) we can eliminate the denominators and obtain the following system:

$$\begin{cases} dx_0 - \sigma(y_0 - \sigma) = \delta x + \tau y \\ \sigma x_0 + d(y_0 - \sigma) = \delta y - \tau x \end{cases} \quad . \tag{22}$$

In order to eliminate τ from this system, we can multiply the first equation by x and add to it the second equation, multiplied by y. We then get:

$$x_0(dx + \sigma y) + (y_0 - \sigma)(dy - \sigma x) = \delta(x^2 + y^2) \quad . \tag{23}$$

We now wish to eliminate σ^2 from the equation and obtain a linear equation in σ. To this end we multiply Equation (18) by x and subtract Equation (23). We obtain:

$$(x - \delta)(x^2 + y^2) = x(x_0^2 + y_0^2 - dx_0) - dy_0 y + \sigma(dy - x_0 y - y_0 x) \quad , \tag{24}$$

so we can now express σ as:

$$\sigma = \frac{(x - \delta)(x^2 + y^2) + (x_0(d - x_0) - y_0^2) + dy_0 y}{(d - x_0)y - y_0 x} \quad . \tag{25}$$

Substituting this expression back into Equation (18) we get:

$$x^2+y^2 = x_0^2 + \left(y_0 - \frac{(x-\delta)(x^2+y^2)+(x_0(d-x_0)-y_0^2)+dy_0y}{(d-x_0)y-y_0x}\right)^2$$

$$\begin{aligned}(x^2+y^2)\left((d-x_0)y-y_0x\right)^2 &= x_0^2\left((d-x_0)y-y_0x\right)^2 + \\ +\big((d-x_0)y_0y-y_0^2x &- (x-\delta)(x^2+y^2)+(y_0^2-x_0(d-x_0))-dy_0y\big)^2\end{aligned}$$

$$\begin{aligned}(x^2+y^2)\left((d-x_0)y-y_0x\right)^2 &= \\ = \underbrace{x_0^2\left((d-x_0)y-y_0x\right)^2}_{\text{I}} &+ \left((x-\delta)(x^2+y^2)+\underbrace{x_0\big((d-x_0)x+y_0y\big)}_{\text{II}}\right)^2 \quad . \end{aligned} \tag{26}$$

One can easily notice that the terms in (26) that do not contain the factor x^2+y^2 all result in $\text{I}+\text{II}^2$, which gives:

$$\begin{aligned} x_0^2(d-x_0)^2y^2 - 2x_0^2(d-x_0)y_0xy + x_0^2y_0^2x^2 &+ \\ +x_0^2(d-x_0)^2x^2 + 2x_0^2(d-x_0)y_0xy + x_0^2y_0^2y^2 &= \\ &= x_0^2\left((d-x_0)^2+y_0^2\right)(x^2+y^2) \quad . \end{aligned}$$

It is therefore possible to reduce Equation (26) by the factor x^2+y^2 to obtain the equation of a quartic curve — that is, an algebraic curve of degree 4:

$$\begin{aligned}\left((d-x_0)y-y_0x\right)^2 &= (x-\delta)^2(x^2+y^2) + \\ &+ 2x_0(x-\delta)\left((d-x_0)x+y_0y\right) + \\ &+ x_0^2\left((d-x_0)^2+y_0^2\right) \quad . \end{aligned} \tag{27}$$

The Special Case

We can now go back to the case where q is an endpoint of L and treat it as a special case of the general case we have just covered. We shall define the coordinate system $\mathcal{C}_R$ such that L lies on its y-axis and q is the origin of $\mathcal{C}_R$ (see Figure 9).

In this case the distance of q from L equals zero, while the same arguments used for the analysis of the general case still hold. We can therefore substitute $d=0$ in Equation (27) and obtain:

$$(x_0y+y_0x)^2 = (x-\delta)^2(x^2+y^2) + 2x_0(x-\delta)(y_0y-x_0x) + x_0^2(x_0^2+y_0^2) \quad . \tag{28}$$

Further Analysis

In Equation (27) we expressed the critical curve in terms of: δ, the distance between the obstacle vertex v and the obstacle edge E; d, which is the distance between the robot vertex q and the half-edge L; and (x_0, y_0) are the coordinates of p under the coordinate system $\mathcal{C}_R$.

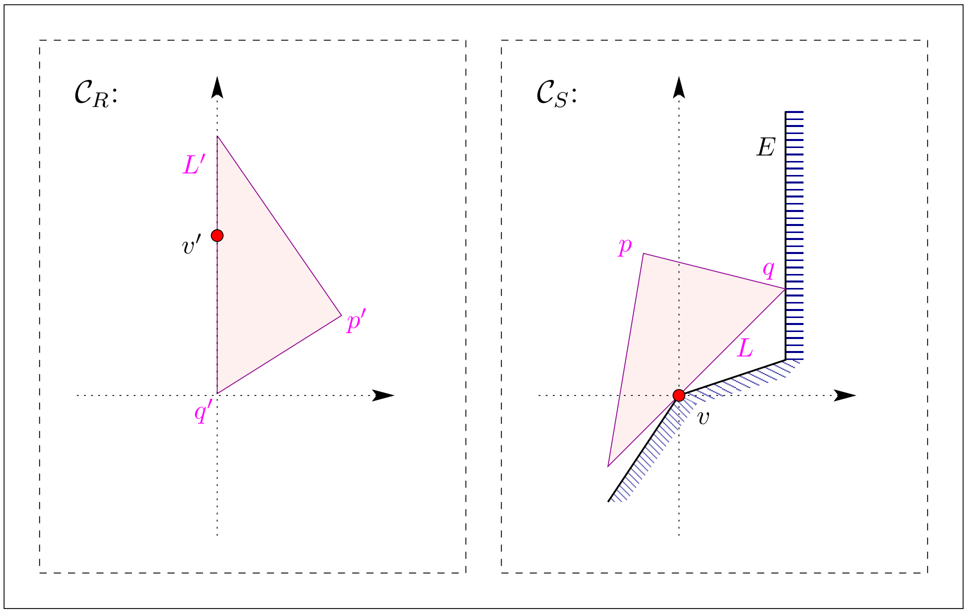

Figure 9: The special case of a curve of type V when q is an endpoint of L.

Let us assume that the endpoints of the robot half-edge L are r_1 and r_2. We can write the distance of p from the line containing L as:

$$d = \frac{|(y_{r_1} - y_{r_2})x_p + (x_{r_2} - x_{r_1})y_p + (x_{r_1}y_{r_2} - x_{r_2}y_{r_1})|}{\sqrt{(y_{r_1} - y_{r_2})^2 + (x_{r_2} - x_{r_1})^2}} \quad . \tag{29}$$

Notice that the denominator of this expression equals $\|L\|$.

Let (x_t, y_t) be the intersection point of a line perpendicular to L going through q with the half-edge L. As we previously showed, this point can be computed from the coordinates of p, r_1 and r_2 and has rational coordinates. To obtain the coordinate system $\mathcal{C}_R$ we first have to translate the coordinates by $(-x_t, -y_t)$ and then rotate it using the rotation matrix $R_{90^\circ - \theta}$, where θ is the angle between L and the x-axis:

$$R_{90^\circ-\theta} \quad = \quad \begin{pmatrix} \sin\theta & \cos\theta \\ -\cos\theta & \sin\theta \end{pmatrix} \quad ,$$

where:

$$\cos\theta = \frac{x_{r_2} - x_{r_1}}{\|L\|} \quad , \qquad \sin\theta = \frac{y_{r_2} - y_{r_1}}{\|L\|} \quad .$$

We thus have:

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = R_{90^\circ-\theta} \begin{pmatrix} x_p - x_t \\ y_p - y_t \end{pmatrix} = \frac{1}{\|L\|} \begin{pmatrix} (y_{r_2} - y_{r_1})(x_p - x_t) + (x_{r_2} - x_{r_1})(y_p - y_t) \\ (x_{r_1} - x_{r_2})(x_p - x_t) + (y_{r_2} - y_{r_1})(y_p - y_t) \end{pmatrix} \quad .$$

It is easy to show that if we substitute d, x_0 and y_0 into Equation (27), we get an equation involving just $\|L\|^2$ and the input coordinates (x_p, y_p), (x_{r_1}, y_{r_1}) and (x_{r_2}, y_{r_2}). It is not difficult to prove a similar result of the special case of Equation (28).

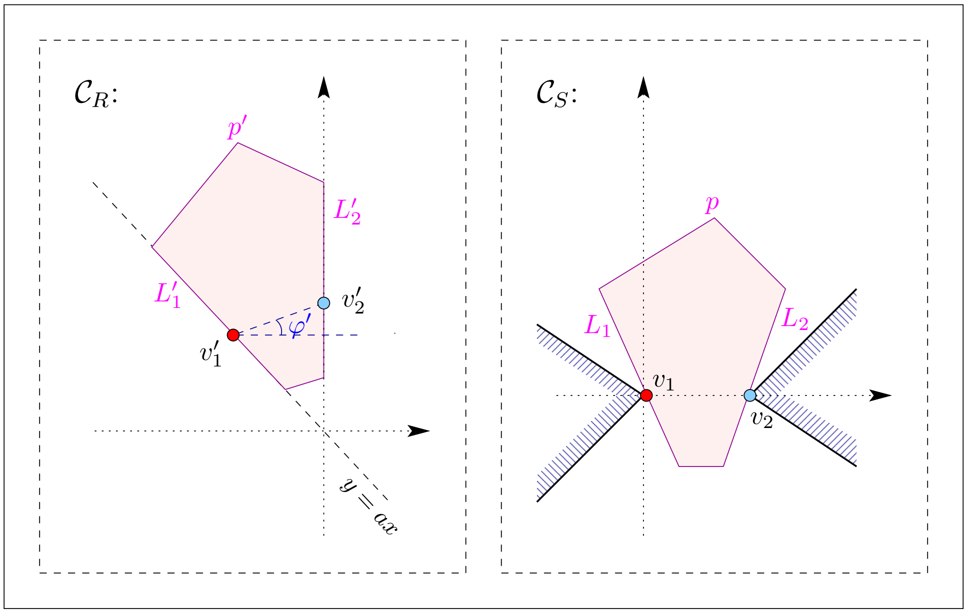

Figure 10: Analysis of a critical curve of type VII.

In an analogous manner, let us assume that the endpoints of the obstacle edge E are u_1 and u_2. The distance between v and the line containing E is:

$$\delta = \frac{|(y_{u_1} - y_{u_2})x_v + (x_{u_2} - x_{u_1})y_v + (x_{u_1}y_{u_2} - x_{u_2}y_{u_1})|}{\sqrt{(y_{u_1} - y_{u_2})^2 + (x_{u_2} - x_{u_1})^2}} \quad . \tag{30}$$

Notice that the denominator of this expression equals $\|E\|$.

We now have to change the coordinate system $\mathcal{C}_R$ to the original coordinate system. We first have to perform rotation using the rotation matrix $R_{\omega-90^\circ}$, where ω is the angle between the edge E and the x-axis in the original coordinate system, and the sine and cosine of this angle obviously have $\|E\|$ as their denominator (see the case of curve of type IV).

Let us examine what happens when we substitute the rotated coordinates and the distance δ into Equation (27). If we consider the monomials in δ, x and y, then we have only monomials of an even degree. As a result, our substitution will yield coefficients that involve only $\|E\|^2$, which is a rational number. We finally translate the origin of $\mathcal{C}_S$ to the location of the vertex $v = (x_v, y_v)$, preserving the "rationality" of the curve coefficients as v has rational coordinates.

4.1.3 Critical Curves of Type VII

We are interested in the location (x, y) of the reference robot vertex p such that the two robot half-edges L_1 and L_2 touch the two convex obstacle vertices v_1 and v_2, respectively.

As we did in our previous curve analyses, let us associate a coordinate system $\mathcal{C}_R$ with the robot, such that L_2 coincides with its y-axis (that is, it lies on the line $x = 0$) and such that L_1 lies on a line that passes through $\mathcal{C}_R$'s origin (that is, it lies on the line $y = ax$, where a is a constant that depends only on the shape of the robot). Let (x_0, y_0) denote the coordinates of the vertex p under $\mathcal{C}_R$. For clarity, we shall denote the relevant robot features under this coordinate system by L'_1, L'_2 and p'.

We also associate another coordinate system $\mathcal{C}_S$ with the obstacles, such that v_1 is its origin and v_2 lies on the x-axis. Thus, if d is the distance between the two vertices, v_2 is located at $(0, d)$ under $\mathcal{C}_S$ (see Figure 10).

Now let us define a rigid transformation $T : \mathcal{C}_R \to \mathcal{C}_S$ in the following manner: Pick a point v_1' on L_1', whose coordinates under $\mathcal{C}_R$ are $(\tau, a\tau)$ and shift the robot such that it coincides with v_1. Then rotate the robot around that point until L_2 touches the obstacle vertex v_2. Let us denote $v_2' = (0, \sigma)$ the point of L_1' which is mapped by T to v_2, and let (x, y) be the location of p under $\mathcal{C}_S$ after this transformation.

The first observation one can make is that since the rigid transformation preserves distances, we have $||p - v_1|| = ||p' - v_1'||$ and $||v_2 - v_1|| = ||v_2' - v_1'||$, hence we can obtain:

$$x^2 + y^2 = (x_0 - \tau)^2 + (y_0 - a\tau)^2 \quad , \tag{31}$$

and:

$$d^2 = \tau^2 + (\sigma - a\tau)^2 \quad . \tag{32}$$

The second important observation is that $\angle p' v_1' v_2' = \angle p v_1 v_2$, since the transformation T does not deform the robot and the angle is preserved. Hence, if we rotate the vector $\vec{v_1' p'}$ by $-\varphi'$ (where φ' is the angle between the vector $\vec{v_1' v_2'}$ and $\mathcal{C}_R$'s x-axis) we will get an identical vector to $\vec{v_1 p}$ (notice that there is no need to further rotate this vector, since $\vec{v_1 v_2}$ is the x-axis itself). That is, we can write:

$$\begin{pmatrix} \cos\varphi' & \sin\varphi' \\ -\sin\varphi' & \cos\varphi' \end{pmatrix} \begin{pmatrix} x_0 - \tau \\ y_0 - a\tau \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} \quad . \tag{33}$$

As the sine and cosine of φ' are given by (recall that $d = \sqrt{\tau^2 + (\sigma - a\tau)^2}$):

$$\cos\varphi' = \frac{-\tau}{\sqrt{\tau^2 + (\sigma - a\tau)^2}} = \frac{-\tau}{d} \quad , \qquad \sin\varphi' = \frac{\sigma - a\tau}{\sqrt{\tau^2 + (\sigma - a\tau)^2}} = \frac{\sigma - a\tau}{d} \quad ,$$

then Equation (33) can be rewritten as:

$$\begin{pmatrix} -\tau & \sigma - a\tau \\ a\tau - \sigma & -\tau \end{pmatrix} \begin{pmatrix} x_0 - \tau \\ y_0 - a\tau \end{pmatrix} = d \begin{pmatrix} x \\ y \end{pmatrix} \quad . \tag{34}$$

Let us now examine the following equation:

$$\begin{pmatrix} x_0 - \tau & y_0 - a\tau \\ a\tau - y_0 & x_0 - \tau \end{pmatrix} \begin{pmatrix} x_0 - \tau \\ y_0 - a\tau \end{pmatrix} = \begin{pmatrix} (x_0 - \tau)^2 + (y_0 - a\tau)^2 \\ 0 \end{pmatrix} \quad .$$

From Equation (31) we can conclude that the vector at the right side of the equation above equals $(x^2 + y^2, 0)$. We can thus multiply Equation (34) by the matrix $\begin{pmatrix} x_0 - \tau & y_0 - a\tau \\ a\tau - y_0 & x_0 - \tau \end{pmatrix}$ and obtain the following (notice that this matrix and the matrix that appear in the left part of Equation (34) both correspond to rotational transformations, hence their multiplication is commutative):

$$\begin{pmatrix} -\tau & \sigma - a\tau \\ a\tau - \sigma & -\tau \end{pmatrix} \begin{pmatrix} x^2 + y^2 \\ 0 \end{pmatrix} = d \begin{pmatrix} x_0 - \tau & y_0 - a\tau \\ a\tau - y_0 & x_0 - \tau \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad , \tag{35}$$

hence we obtain the following system of linear equations:

$$\begin{cases} -\tau(x^2 + y^2) = d(x_0 x + y_0) - d(x + ay)\tau \\ (a\tau - \sigma)(x^2 + y^2) = d(x_0 y - y_0 x) + d(ax - y)\tau \end{cases} \quad . \tag{36}$$

Notice that σ does not appear in first equation, and we can readily obtain an expression for τ:

$$\tau = \frac{d(x_0 x + y_0 y)}{d(x + ay) - (x^2 + y^2)} \quad . \tag{37}$$

Substituting τ back into Equation (31) we get:

$$x^2 + y^2 = \left(x_0 - \frac{d(x_0 x + y_0 y)}{d(x + ay) - (x^2 + y^2)}\right)^2 + \left(y_0 - \frac{ad(x_0 x + y_0 y)}{d(x + ay) - (x^2 + y^2)}\right)^2$$

$$\begin{aligned} &(x^2 + y^2)\left(d(x + ay) - (x^2 + y^2)\right)^2 = \\ &= \left(\underbrace{d(ax_0 - y_0)y}_{\mathrm{I}} - x_0(x^2 + y^2)\right)^2 + \left(\underbrace{d(y_0 - ax_0)x}_{\mathrm{II}} - y_0(x^2 + y^2)\right)^2 \quad . \end{aligned} \tag{38}$$

As we did in the previous section for curves of type V, we shall examine all the terms of Equation (38) that do not contain the factor $x^2 + y^2$. It is easy to see that they all result in $\mathrm{I}^2 + \mathrm{II}^2$, which gives:

$$(d(ax_0 - y_0)y)^2 + (d(y_0 - ax_0)x)^2 = d^2(ax_0 - y_0)^2(x^2 + y^2) \quad .$$

It is therefore possible to reduce Equation (38) by the factor $x^2 + y^2$ and obtain the equation of a quartic curve:

$$\begin{aligned} \left(d(x + ay) - (x^2 + y^2)\right)^2 &= (x_0^2 + y_0^2)^2(x^2 + y^2) + \\ &+ 2d(ax_0 - y_0)(y_0 x - x_0 y) + \\ &+ d^2(ax_0 - y_0)^2 \quad . \end{aligned} \tag{39}$$

Further Analysis

We shall now transform the curve given in Equation (39) to the original coordinate system, using similar techniques those used for curves of type IV and V.

We first note that the slope a depends on the tangent of the angle between the two supporting lines of L_1 and L_2, and is a rational expression that involves the coordinates of the endpoints of the two half-edges (see Equation (30) for a similar expression).

Moreover, if we denote the intersection between the supporting lines lines of L_1 and L_2 by (x_t, y_t), we can express x_0 and y_0 using the the following expression, where r_1 and r_2 denote the endpoints of L_1 and θ is the angle its supporting line forms with the x-axis:

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = R_{90^\circ - \theta} \begin{pmatrix} x_p - x_t \\ y_p - y_t \end{pmatrix} = \frac{1}{\|L_1\|} \begin{pmatrix} (y_{r_2} - y_{r_1})(x_p - x_t) + (x_{r_2} - x_{r_1})(y_p - y_t) \\ (x_{r_1} - x_{r_2})(x_p - x_t) + (y_{r_2} - y_{r_1})(y_p - y_t) \end{pmatrix} \quad .$$

Note that all monomials in x_0 and y_0 of Equation (39) are of an even degree, so if we substitute the expressions for x_0 and y_0 we obtained above into this equation, our curve equation will only contain $\|L_1\|^2$, which is a rational number.

We finally apply a rigid transformation to convert $\mathcal{C}_S$ to our original coordinate system. To do this, we first rotate by ω, which is the angle between the vector $\vec{v_1 v_2}$ and the x-axis in the original coordinate system. Clearly:

$$\cos\omega = \frac{x_{v_2} - x_{v_1}}{\sqrt{(x_{v_2} - x_{v_1})^2 + (y_{v_2} - y_{v_1})^2}} = \frac{x_{v_2} - x_{v_1}}{d} \quad , \qquad \sin\omega = \frac{y_{v_2} - y_{v_1}}{d} \quad .$$

As all monomials in x, y and d in of Equation (39) are of an even degree, this results coefficients that only involve d^2, which is a rational number. We finally translate the origin to v_1, and obtain a curve of degree 4 with rational coefficients.

4.2 The Combinatorial Construction

Having computed the set of critical curves for the given robot and obstacles, we next have to construct the planar arrangement of these curves. As we did for the case of a ladder, we shall associate a *characteristic label* to each arrangement cell, but we will have to extend our definitions a bit.

Having located the reference vertex p at some position within a certain cell C of the arrangement of critical curves, certain orientations will be collision-free, while others will cause an interference of the robot with the obstacles, thus will be forbidden. The *critical angles* that separate between ranges of free and forbidden orientations are characterized by when either of the following events:

- A robot vertex q_i touches an obstacle edge E_j.
- An obstacle vertex v_j touches a robot half-edge L_i.

Thus, we can label each critical angle with a pair of obstacle feature (either a vertex or an edge) and robot feature (either a vertex or a half-edge) that come into contact. Since each pair of critical angles define a range of free orientations for the robot, we can add an interval of two such pairs to the characteristic label of C (and if all the orientations within a cell are free, we will use the notation $\chi(C) = \big\{[\langle\Omega,\Omega\rangle, \langle\Omega,\Omega\rangle]\big\}$, where Ω stands for a non-existing feature). The definition of the critical curves ensures that this characteristic label is a topological property of the entire arrangement cell.

For example, in Figure 3 the characteristic label of the cell that the current position of p belongs to is $\big\{[\langle E_1, q_1\rangle, \langle v_1, L_3\rangle],\ [\langle E_3, q_1\rangle, \langle E_4, q_1\rangle],\ [\langle E_4, q_1\rangle, \langle E_1, q_2\rangle]\big\}$.

We can now define three-dimensional cells in an analogous way to what we did in the ladder case: Given a position $(x, y) \in C$ and $[\mathcal{F}_1, \mathcal{F}_2] \in \chi(C)$, let us denote the two critical angles related to the two feature pairs $\mathcal{F}_1, \mathcal{F}_2$ by $\lambda_l^{(C)}(x, y, \mathcal{F}_1)$ and $\lambda_h^{(C)}(x, y, \mathcal{F}_2)$. We shall also use the convention $\lambda_l(x, y, \langle\Omega,\Omega\rangle) = 0°$ and $\lambda_h(x, y, \langle\Omega,\Omega\rangle) = 360°$ for any cell. We can now define a three-dimensional free cell κ of as:

$$\kappa = \text{cell}(C, \mathcal{F}_1, \mathcal{F}_2) = \left\{\ \langle x, y, \theta\rangle \ \mid\ (x, y) \in C, \ \ \theta \in \left(\lambda_l^{(C)}(x, y, \mathcal{F}_1), \lambda_h^{(C)}(x, y, \mathcal{F}_2)\right)\ \right\}$$

We shall now construct a graph $\mathcal{G}$ with the three-dimensional cells of free configurations as its vertices. For each neighboring two-dimensional cells C and C' we shall do the following:

- For each $[\mathcal{F}_1, \mathcal{F}_2] \in \chi(C) \cap \chi(C')$ we shall connect cell$(C, \mathcal{F}_1, \mathcal{F}_2)$ and cell$(C', \mathcal{F}_1, \mathcal{F}_2)$.
- For each $[\mathcal{F}_1, \mathcal{F}_2] \in \chi(C) \setminus \chi(C')$ we shall connect cell$(C, \mathcal{F}_1, \mathcal{F}_2)$ with all other cells $\kappa' =$ cell$(C', \mathcal{F}_1', \mathcal{F}_2')$ that are induced by C'.

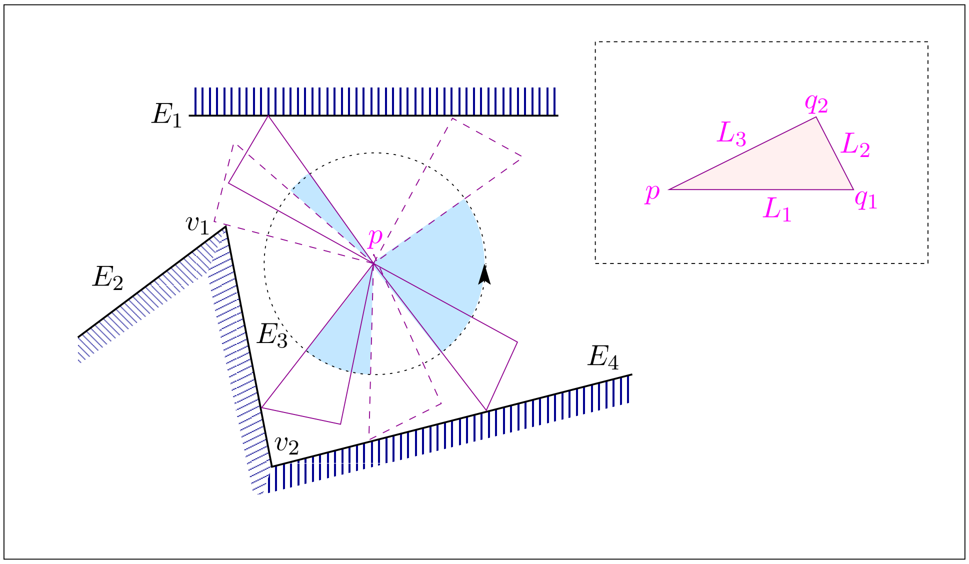

Figure 11: The legal angle ranges formed by L_1 and the x-axis for a given placement of reference vertex p (shaded). The critical orientations of the robot are also drawn, where clockwise stops are drawn as solid polygons and counterclockwise stops are drawn as dashed polygons. Notice the obstacle and robot features (vertices and edges) that define the critical angles.

- For each $[\mathcal{F}'_1, \mathcal{F}'_2] \in \chi(C') \setminus \chi(C)$ we shall connect cell$(C', \mathcal{F}'_1, \mathcal{F}'_2)$ with all other cells $\kappa =$ cell$(C, \mathcal{F}_1, \mathcal{F}_2)$ that are induced by C.

We are now ready to answer motion-planning queries using the connectivity graph $\mathcal{G}$. Given a start configuration c_s and a goal configuration c_t, we locate the two cells κ_s and κ_t that contain these two configurations and check whether they are connected in $\mathcal{G}$. If not, we have a combinatorial disconnection proof. Otherwise, we can trace one of the paths that connect the two cells are devise a collision-free motion for our polygonal robot.

5 Conclusions

We have shown that all critical curves that occur in the piano-movers' algorithms can be represented as algebraic curves of degree 4 at most with rational coefficients, assuming that the input consists of rational numbers. This opens the way for an implementation of a traits class for CGAL's arrangement package that can handle all geometric predicates and constructions necessary for the robust construction of the arrangement of these curves.

One should note that if the input obstacles have a total of n vertices, then there we may have $O(n^2)$ critical curves in case of a ladder, and $O(n^2m^2)$ critical curves in case of a polygonal robot with m vertices. The complexity of the resulting arrangement can thus become $O(n^4)$ and $O(n^4m^4)$, respectively. As exact computation with algebraic numbers can slow down the performance of the geometric algorithms by 2–3 orders of magnitude, in comparison to computation with machine-precision floating-point numbers, it seems that handling large-scale problems using the piano-mover's algorithm would be highly inefficient.

However, as exact computations are required only in degenerate or nearly-degenerate situations, it may be the case that most parts of the arrangement can be created using a floating-point representation, while the exact computations will only be invoked in the more "problematic" cases, where floating-point arithmetic yields ambiguous results.

References

[1] The CGAL project homepage. `http://www.cgal.org/`.

[2] The CORE library homepage. `http://www.cs.nyu.edu/exact/core/`.

[3] The LEDA homepage. `http://www.algorithmic-solutions.com/enleda.htm`.

[4] F. Avnaim, J.-D. Boissonnat, and B. Faverjon. A practical exact motion planning algorithm for polygonal object amidst polygonal obstacles. In *Geometry and Robotics*, volume 391 of *Lecture Notes in Computer Science*, pages 67–86. Springer-Verlag, 1989.

[5] J. Bañon. Implementation and extension of the ladder algorithm. In *Proc. IEEE Conf. on Robotics and Automation*, pages 1548–1553, 1990.

[6] C. Burnikel, S. Funke, K. Mehlhorn, S. Schirra, and S. Schmitt. A separation bound for real algebraic expressions. In *Proc. 9th European Symposium on Algorithms (ESA 01)*, pages 254–265, 2001.

[7] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties and Algorithms*. Springer-Verlag, 1992.

[8] E. Fogel, R. Wein, and D. Halperin. Code flexibility and program efficiency by genericity: Improving CGAL's arrangements. In *Proc. 12th European Symposium on Algorithms (ESA 04)*, pages 664–676. Springer-Verlag, 2004.

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

[10] C. Li and C. Yap. New constructive root bound for algebraic expressions. In *Proc. 12th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2001.

[11] J. T. Schwartz and M. Sharir. On the "piano movers" problem I: The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. Technical Report 39, Department of Computer Science, New-York Univ., New-York, NY, October 1981.

[12] J. T. Schwartz and M. Sharir. On the "piano movers" problem I: The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. *Commun. Pure Appl. Math.*, 36:345–398, 1983.

[13] R. Wein, J. P. van den Berg, and D. Halperin. The Visibility–Voronoi complex and its applications, 2004. `http://www.cs.tau.ac.il/~wein/publications/pdfs/vv_complex_TR.pdf`.