Planning Motions of Polyhedral Parts by Rolling

Alessia Marigo, Antonio Bicchi

Keywords— **Discrete nonholonomic systems, Robotic manipulation, Group theory applications.**

Abstract

The nonholonomic nature of rolling between rigid bodies can be exploited to achieve dextrous manipulation of industrial parts with minimally complex robotic effectors. While for parts with smooth surfaces a relatively well developed theory exists, planning for parts with only piece–wise smooth surfaces is largely an open problem.

The problem of arbitrarily displacing and reorienting a polyhedron by means of rotations about edges belonging to a fixed plane is considered. Relevant theoretical results are reviewed, and a polynomial time algorithm is proposed that allows planning such motions. The effects of finite accuracy in representing problem data, as well as the operational and computational complexity of the method are considered in detail.

I. Introduction

A recent trend of research in Robotics tends to meet industry needs of economy and reliability in the design of manipulation devices, by investigating minimally complex hardware systems for a given manipulation task (see e.g. [1]). To this end, the nonholonomic behaviour of some systems has been exploited to achieve dexterous manipulation with simple mechanical design. In fact, one of the characteristics of nonholonomic systems is that, by means of cyclic paths of some of their state variables, a controlled change in other variables can be produced. Bicchi and Sorrentino [2] designed and implemented a dexterous hand using only three motors. Such hand is able to arbitrarily change the position and orientation of the manipulated object by rolling it between the fingertips, provided that its surface is convex and regular (see [3]).

In order to approach more genuinely industrial problems, in this paper we consider a similar style of manipulation as applied to parts with only piece–wise regular surface, and particularly polyhedral parts. The rolling of a polyhedron on a plane is itself a nonholonomic phenomenon, as it can be checked by rolling a die on a table along cyclic paths (see e.g. fig. I). However, a completely different set of tools are necessary to analyze and plan rolling motions of polyhedral parts than those of regular surfaces, the latter being mostly a problem of differential geometry, while the former is intermingled with discrete mathematics.

The work has been conducted with partial support of NATO CR grant no. 960750 and ASI ARS-96-170.

The authors are with Centro Interdipartimentale di Ricerca "Enrico Piaggio" of the University of Pisa, 56100 Pisa, Italy. E–mail: marigo@piaggio.ccii.unipi.it, bicchi@ing.unipi.it.

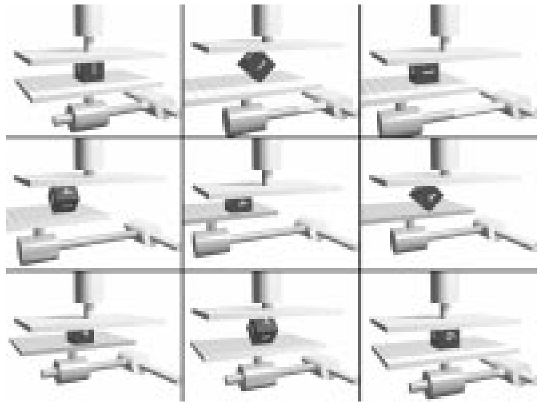


Fig. 1. A parallel–jaw gripper with three d.o.f. can manipulate polyhedral parts. Plates are covered by compliant, high–friction pads. Vertical motion is passively controlled so as to accomodate for the changing height of parts while keeping suitable grasping forces.

Previous work on graspless manipulation of polyhedral parts by rolling in the robotics literature include that of Sawasaki, Inoue and Inaba [4], Inoue and Aiyama [5], and Erdmann and Mason [6]. In [7], a theoretical analysis of the set of configurations a polyhedron can reach by rolling was presented. Such analysis showed that such set may have a rather complex structure, exhibiting radically different behaviour (such as density or discreteness in the configuration space), depending in a very sensitive fashion on the geometric parameters of the polyhedron.

In this paper we tackle the problem of manipulating polyhedral parts by rolling from a more application–oriented viewpoint. Firstly, we want to provide an algorithm for planning such manipulation, i.e., given an initial and final configuration pair, to find a sequence of simple rolling motions that brings the part from the former to the latter. Given that not always such path may exist, and the sensitivity of the existence conditions to part parameters, we must study this problem in connection with *tolerances* on parameters and specifications. In order for this style of manipulation to be viable in practice, bounds on the number of rolling motions necessary to reach the goal configuration to within a prescribed error must be provided. We call the study of such bounds operational complexity analysis. On the other hand, a computational complexity analysis is also performed in terms of time occupation by the planning algorithm.

II. Theoretical Analysis

A. Background

Manipulated parts are considered that have a piecewise flat, closed surface, comprised of a finite number of faces, edges, and vertices. Observe that actual parts need not be convex, in general. However, the finger plates being assumed to be large w.r.t. the diameter of parts, we need only be concerned with the convex hull of parts themselves.

We also restrict to motions of a polyhedron on a plane which are given by sequences of rotations about one of the edges in contact, by the amount that exactly brings an adjacent face to ground. This action on the parts, which will be referred to as an elementary tumble (tumble for short), appears to be more reliable than slipping or pivoting about the vertices. By affixing labels to the faces of the polyhedron, a sequence of tumbles can be described by the sequence of labels of the faces in turn contacting the plane. Observe that not all the sequences of indices are admissible as any two consecutive labels must correspond to adjacent faces.

The configuration space of a polyhedron rolling on a plane is denoted by $\widetilde{M} = \mathbb{R}^2 \times S^1 \times \{F_1, ..., F_l\}$, where the set $\{F_1, ..., F_l\}$ is the set of faces of the polyhedron and S^1 is the unit sphere of dimension 1. Thus a point $(x, y, \theta, F_i) \in \widetilde{M}$ describes the face F_i in contact with the plane, the angle $\theta \in S^1$ between two reference systems fixed respectively on the plane and on face F_i, and $(x, y) \in \mathbb{R}^2$ are the coordinates of the origin of the reference system fixed on face F_i. Observe that the state space is the union of l copies of $\mathbb{R}^2 \times S^1$. The subset of reachable configurations from some initial configuration (which, without loss of generality, will be henceforth taken as $(0, 0, 0, F_1)$), is given by the set of points reached by applying all admissible sequences of tumbles to the initial configuration. We will denote by $\mathcal{R}_1$ the reachable set.

Notice that the set of all sequences is an infinite but countable set while the configuration space is a finite disjoint union of copies of a 3–dimensional variety. Thus, the set of reachable points is itself countable. Therefore, instead of the more familiar concept of "complete reachability" (corresponding to $\mathcal{R}_1 = \widetilde{M}$) , it will only make sense to investigate a property of "dense reachability" defined as $\mathsf{closure}(\mathcal{R}_1) = \widetilde{M}$. In other words, rolling a polyhedron on a plane has the dense reachability property if, for any configuration of the polyhedron and every $\epsilon \in \mathbb{R}_+$, there exist a finite sequence of tumbles that brings the polyhedron closer to the desired configuration than ϵ. We refer in particular to a distance on $\widetilde{M}$ defined as

$$\|(x_1, y_1, \theta_1, F_i) - (x_2, y_2, \theta_2, F_j)\| = \\ = \max\left\{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2},\ |\theta_1 - \theta_2|,\ 1 - \delta(F_i, F_j)\right\}.$$

The following notation is introduced for further convenience:

Let $\mathcal{P}$ denote the convex polyhedron rolling on a plane P, and let $\mathcal{V} = \{V_1, \ldots, V_m\}$ be the set of vertices, $\mathcal{E} = \{E_1, \ldots, E_k\}$ the set of edges, and $\mathcal{F} = \{F_1, \ldots, F_l\}$ the set of faces of $\mathcal{P}$. Let D_{lk} denote the length of the edge incident to vertices V_l and V_k. Also, for each face F_j and each vertex V_i belonging to such face denote α_{ij} the angle between the two edges on face F_j incident at vertex V_i. of face F_j at vertex V_i. For each vertex V_i, the *defect angle* β_i is defined as the complement to 2π of the sum of angles α_{ij} for all j such that face F_j is adjacent to V_i, i.e. $\beta_i = 2\pi - \sum_j \alpha_{ij}$.

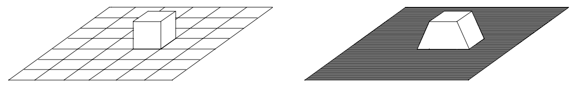


Fig. 2. Slightly different polyhedra may reach dramatically different sets of configurations by rolling

The geometric structure of the reachable set for a polyhedron rolling on a plane has been thoroughly investigated in [8], where the following theorem is proved:

Theorem 1: The set of reachable configurations for a polyhedron rolling about its edges is dense in $\widetilde{M}$ if and only if there exists a vertex V_i of the polyhedron whose defect angle is irrational with π, i.e., iff $\exists \beta_i : \frac{\beta_i}{\pi} \notin Q$.

Remark 1. Observe that in the above reachability theorem the conditions upon which the density or discreteness of the reachable set depends are in terms of rationality of certain parameters and their ratios. This entails that two very similar polyhedra may have qualitatively different reachable sets. This is for instance the case of fig. 2, where the projection on $\mathbb{R}^2$ of the reachable set is illustrated for a unit cube (giving a square lattice), and for a truncated pyramid obtained from the cube by slightly shrinking its upper face, which gives a dense set. In fact, for any polyhedron whose reachable set has a discrete structure, there exists an arbitrarily small perturbation of some of its geometric parameters that achieves density. In view of this remark, and considering that in applications the geometric parameters of the parts will only be known to within some *tolerance*, i.e., a bounded neighborohood of their nominal value, a formulation of the planning problems such as

"Given a polyhedral part $\mathcal{P}$ and a final configuration C_f, find a sequence of tumbles that brings $\mathcal{P}$ in C_f"

is clearly ill-posed.

To provide a correct model of the phenomenon of rolling polyhdral parts, it is necessary to deal with the representation of toleranced quantities in the computer. A certain parameter of nominal value $\hat{a}$ is given with tolerance τ_a, and written $\hat{a} \pm \tau_a$, if its true (unknown) value a satisfies

$$\hat{a} - \tau_a \le a \le \hat{a} + \tau_a.$$

In this paper, for the number a we will consider the rational representation $\bar{a} = \frac{p_a}{q_a}$ with p_a, q_a integers, such that $|\bar{a} - \hat{a}| \le \tau_a$, can be found by using the continued fraction expansion of $\hat{a}$ which is recalled here for convenience:

$$\bar{a} = \frac{p_a}{q_a} = n_0 + \cfrac{1}{n_1 + \cfrac{1}{n_2 + \cfrac{1}{\cdots + \frac{1}{n_N}}}}$$

where n_i are positive integers such that n_0 is the inferior integer part of $\hat{a}$, $n_0 = \lfloor \hat{a} \rfloor$, $n_1 = \lfloor \frac{1}{\hat{a} - n_0} \rfloor$, $n_2 = \lfloor \frac{1}{\frac{1}{\hat{a} - n_0} - n_1} \rfloor$, and so on. For such truncated representation, it holds

$$|\hat{a} - \frac{p_a}{q_a}| \leq \frac{1}{q_a^2}.$$

Therefore, a representation of $a \pm \tau_a$ is given by a truncated continued fraction expansion $\bar{a} = p_a/q_a$ such that $q_a \geq {\tau_a}^{-1/2}$. In order to avoid nonsensical excess in representation accuracy, we assume $q_a = \lceil {\tau_a}^{-1/2} \rceil$ (this is obtained in at most $N = q_a$ steps). For an angular parameter the same representation applies, proviso angles are measured in π rad units.

As a consequence of such rational representation of polyhedra, the structure of the reachable set will always be discrete. The study of the lattice of discrete reachable sets is therefore crucial to the developments of this paper. As a final observation, it must be pointed out that in the execution of a rolling manipulation plan, the actual configuration finally reached by the polyhedron will depend on the true values of the geometric parameters, and hence will differ from the planned one by an error propagating through manipulation.

B. Problem Statement and Method of Solution

In view of the above discussion, a correct formulation of the planning problem can be given as follows:

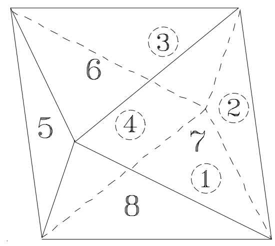
The Planning Problem. *Given a polyhedron $\mathcal{P}$ with known geometric parameters and tolerances, a desired final configuration C_f, and an accuracy ϵ, decide whether a sequence of tumbles reaching a configuration $\hat{C}_f$ such that $\|\hat{C}_f - C_f\| \leq \epsilon$ exists; if so, provide one such sequence.*

A brute force approcah to the solution of the planning problem is a graph search. The configuration graph size for sequences of length N is $l(l-1)^{N-1}$. The complexity of the search can not be bounded as a bound on N can not be given *a priori*.

The method we propose exploits a particular group structure that can be recognized in a subset of all manipulation sequences. The generators of such group will be used to provide a definite answer to the solvability part of the planning problem, and to provide a solution if one exists. A bound on the length of the solution sequence will also be provided, although no optimality claim is made on the proposed solution. The fundamentals of such group–theoretical technique are described in this section.

Denote by $\mathcal{P}_D$ a plane development of the polyhedron $\mathcal{P}$ on the plane, i.e. any planar connected union of l polygons, each one a copy of a distinct face of $\mathcal{P}$, such that two polygons are adjacent only if the corresponding faces of the polyhedron are adjacent. The copy of the reference face F_1 coincides with F_1.

Label the polygons with the same indices of the corresponding faces of the polyhedron. Define for each $i \neq 1$ the sequence L_i of indices such that L_i is a path on $\mathcal{P}_D$ from polygon F_1 to polygon F_i through adjacent polygons.



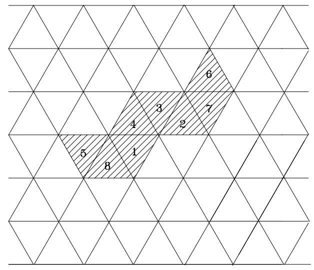


Fig. 3. An octahedron and its development on the plane. Circled numbers refer to hidden faces.

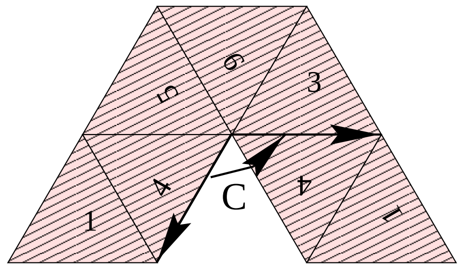


Fig. 4. For the development on the plane of a octahedron, the sequence $(1, 4, 5, 6, 3, 4, 1)$ produces a rigid rotation of an angle $2\pi/3$ about point C.

We call $\{L_i,\ i = 2, ..., l\}$ the set of *transit paths.* Observe that a transit path L_i can be naturally associated with a sequence of tumbles of the polyhedron that brings face F_i in contact with the plane P.

Consider a vertex V_j of $\mathcal{P}$. Let $F_{j_1}, F_{j_2}, ..., F_{j_{k_j}}$ be all the faces of $\mathcal{P}$ sharing vertex V_j. Up to reordering labels, we can suppose F_{j_r} and $F_{j_{r+1}}$ are adjacent for all $r = 1, ..., k_{j-1}$ and F_{j_1} is adjacent to $F_{j_{k_j}}$. Consider the polygon in $\mathcal{P}_D$ corresponding to F_{j_1}, and let C_j be the vertex on it corresponding to V_j. Consider for all $j = 1, ..., m$ the sequences $S_j = (j_1, j_2, ..., j_{k_j}, j_1)$ and $R_j = L_{j_1} S_j L_{j_1}^{-1}$, where by L^{-1} we mean the sequence L followed inversely. In the sequel, we will occasionally refer to such composition of sequences of tumbles as a concatenation of strings. Observe that the new string R_j produces a rigid rotation of the polyhedron by an angle β_j about an axis through C_j and perpendicular to P (fig.II-B).

The set of all admissible sequences starting and ending with face F_1 in contact with the plane, together with a composition law given by concatenation of sequences, forms a group $\mathbf{R}$ modulo the equivalence relation wich is defined below. Any sequence in $\mathbf{R}$ can be written in its reduced form by eliminating trivial (palindromic) substrings, corresponding to vane manipulations, which may appear in the sequence. Any two sequences are equivalent if their reduced form are equal. It can be proved (see [8]) that a finite set of generators for $\mathbf{R}$ is given by $R = \{R_j,\ j = 1, ..., m-1\}$. The group $\mathbf{R}$ is therefore referred to as the group of planar rotations of the polyhedron.

Recalling that all angles are represented as rationals in π radians unit, the defect angles of $\mathcal{P}$ are given (in radians) by $\beta_i = \frac{p_i}{q_i}\pi$. We define the greatest common divisor of the defect angles as $\bar{\beta} = \text{GCD}(\beta_j,\ j = 1, ..., m-1) = \frac{p}{q}\pi$. By an iterated version of the extended Euclidean algorithm [9] we can then find integers n_j such that

$$\bar{\beta} = \sum_{j=1}^{m-1} n_j \beta_j \pmod{2\pi}.$$

Also define the angle $\beta = \frac{\pi}{q}$. Observe that the smallest angle of rotation, i.e. the smallest feasible change of orientation, is β if p is odd and 2β if p is even. The element of the group of planar rotations that performs a rotation (about a point $\bar{C}$) of $\bar{\beta}$ is $\bar{R} = \prod_{j=1}^{m-1} R_j^{n_j}$. The element that produces the smallest possible rotation has the form $\bar{R}^k$, with $k \leq q$ for p odd or $k \leq 2q$ for p even.

Consider now the set of all sequences starting and ending with face F_1 in contact with the plane such that the final configuration of polyhedron is only translated. This set, with the composition by concatenation, forms a subgroup $\mathbf{T} \subset \mathbf{R}$. It can be proved (see [8]) that a finite set of generators for $\mathbf{T}$ is given by

$$T = \{t_{jk} = \bar{R}^{-k}\left(\bar{R}^{h_j} R_j^{-1}\right)\bar{R}^k \ ;\ j = 1, ..., m-1; \quad k \in Z \text{ such that } -q+1 \leq k \leq q-1\}. \tag{1}$$

where h_j are the integers such that $h_j\bar{\beta} = \beta_j$. The group $\mathbf{T}$ is referred to as the group of translations of the polyhedron. Translations t_{jk} correspond to strings of tumbles, whose trivial substrings have been purged.

Remark 2. For all $j = 1, ..., m-1$, the generators t_{jk} can be seen as $2q-1$ copies of t_{j0}, rotated of β with respect to each other.

III. Planning Algorithm

The model of the polyhedral part is stored in the form of a list of faces whose items contain a list of the edges of the face. Each item of the edge list in turn contains a pair of vertices. Finally, each element of these pairs points to a three dimensional vector containing the cartesian coordinates of the corresponding vertex. The defect angles $\beta_i = \frac{p_i}{q_i}\pi$ are then computed for each vertex, and their value attached to the vertex list.

The polyhedron is developed on the plane by the following steps:
Step 1. Place the polygon corresponding to F_1 in position $(0, 0)$ with orientation angle 0;
Step 2. If l polygons have been drawn then stop, else draw a new polygon adjacent to one already drawn (see fig.II-B), and repeat Step 2.

One fundamental step of our algorithm is the computation of the angle $\bar{\beta}$ by the formula

$$\bar{\beta}/\pi = \frac{p}{q} = \text{GCD}(\frac{p_i}{q_i};\ i = 1, ..., m-1)$$

and the computation of a set of integers $n_i = 1, ..., m$ such that

$$\frac{p}{q} = \sum_{i=1}^{m-1} n_i \frac{p_i}{q_i}. \tag{2}$$

Although finding the smallest such n_j's would be interesting for the sake of reducing the amount of tumbles to obtain a given rotation, this problem is notoriously a hard one in integer programming, which may add significantly to the algorithm complexity. Instead, we find a set of integers by applying the Extended Euclidean Algorithm (EEA), and take their values modulo $2q_j$ (otherwise there would exist $n'_j \leq 2q_j$ such that $n_j \frac{p_j}{q_j}\pi \equiv n'_j \frac{p_j}{q_j}\pi \pmod{2\pi}$). By these means, we also provide a bound on the n_j's.

These computations are done by using the well–known Euclidean Algorithm (EA) and its Extended version (EEA). To find $p = \text{GCD}(p_i\bar{q}_i,\ i = 1, ..., m-1)$, where $\bar{q}_i = \frac{q}{q_i}$, $q = \text{lcm}(q_i;\ i = 1, ..., m-1)$, the basic EA step finding the GCD of two integers is applied recursively as

$$p = \text{GCD}(p_{m-1}\bar{q}_{m-1}, \text{GCD}(p_{m-2}\bar{q}_{m-2}, \ldots \text{GCD}(p_3\bar{q}_3, \text{GCD}(p_2\bar{q}_2, p_1\bar{q}_1))\ldots))$$

The EEA is first applied to find coefficients (n_2^0, n_1^0) solving

$$\text{GCD}(p_2\bar{q}_2, p_1\bar{q}_1) = n_2^0\, p_2\bar{q}_2 + n_0^1\, p_1\bar{q}_1,$$

then to find coefficients (n_3^1, n_2^1) for the pair

$$(p_3\bar{q}_3, \text{GCD}(p_2\bar{q}_2, p_1\bar{q}_1)).$$

The triple of coefficients $(n_3^1, n_2^1 n_2^0, n_2^1 n_1^0)$ is thus obtained solving

$$\text{GCD}(p_3\bar{q}_3, p_2\bar{q}_2, p_1\bar{q}_1) = n_3^1\, p_3\bar{q}_3 + n_2^1 n_2^0\, p_2\bar{q}_2 + n_2^1 n_0^1\, p_1\bar{q}_1.$$

By iterating this procedure, we get the $m-1$ coefficients n_i as

$$\begin{array}{rcl} n_{m-1} & = & n_{m-1}^{m-3}, \\ n_{m-2} & = & n_{m-2}^{m-3} n_{m-2}^{m-4}, \\ n_{m-3} & = & n_{m-2}^{m-3} n_{m-3}^{m-4} n_{m-3}^{m-5}, \\ \vdots & & \\ n_1 & = & \prod_{i=1}^{m-1} n_i^{i-1} \end{array}$$

Next fundamental part of the algorithm is the computation of the set T of generators of translations, defined in eq.1. By some simple computations, the analytic expression for the generators is obtained as

$$t_{jk} = (C_j - \bar{C})(e^{i\beta_j} - 1)e^{ik\bar{\beta}}$$

for $j = 1, ..., m-1$ and $k \in Z$such that $-q+1 \leq\ k \leq q-1$, where point $\bar{C}$ is given by

$$\bar{C} = \prod_{j=1}^{m-1} R_j^{n_j} e^{-i\bar{\beta}}.$$

Remark 3. The group of possible translations of the polyhedron is given by all integer combinations of the generators t_{jk}. To simplify notation, we will denote henceforth these generators by a single index, i.e. let $T = \{v_1, \ldots, v_N\}$, $N = 2q(m-1)$. Assume that v_i, v_j are the two linearly independent generators. Any generator v_k can be expressed as the sum of its components along v_i and v_j,

$$v_k = a_{ki}v_i + a_{kj}v_j,$$

with $[a_{ki}\, a_{kj}]^T = [v_i\, v_j]^{-1} v_k$. The span of v_i, v_j, v_k over the integers can be more efficiently represented as the span of v_1, v_2 over the smallest field (Q or $\mathbb{R}$, $Q \subset \mathbb{R}$) where a_{ki} and a_{kj} take values, respectively. The group generated by vectors in T is therefore dense in $\mathbb{R}^2$ if and only if there exists a triple of vectors v_i, v_j, v_k such that both $a_{ki} \in \mathbb{R} \setminus Q$ and $a_{kj} \in \mathbb{R} \setminus Q$; discrete if and only if for all triples v_i, v_j, v_k, both $a_{ki} \in Q$ and $a_{kj} \in Q$. The group will consist of the union of a countable number of disjoint dense monodimensional subspaces if neither condition applies.

In the same spirit of the observation made in Remark 1, even when the density condition of Remark 3 is verified, it will be most expedient to replace the original generators v_k by an approximation $\tilde{v}_k$ such that

$$\tilde{v}_k = \frac{r_k^i}{s_k^i} v_i + \frac{r_k^j}{s_k^j} v_j$$

where the rational approximations of the real coefficients are obtained through use of the continued fraction expansion such that

$$\left| \frac{r_k^l}{s_k^l} - a_{kl} \right| \leq \tau_g \quad l = i, j. \tag{3}$$

Representing all generators in terms of two of them by rational coefficients entails that the group of translations of the polyhedron is reduced to the set of integer combinations of the two basic generators, i.e. a discrete set, namely a two–dimensional lattice. Notice that, if the original group is dense, the lattice mesh can be chosen as small as desired by setting a small bound τ_g on the approximation error. If the original group is discrete, an approximation can still be introduced of a rational coefficient with another one, whose denominator is smaller. Such lattice approximation of the actual set of reachable positions (the accuracy of which is under control by the parameter τ_g) is one key idea in the solution we give to the solvability part of the planning problem, and is also instrumental in devising an efficient planner.

Motivated by the arguments in Remark 3, we consider henceforth a lattice structure in the group $\mathbf{T}$, generated by $\tilde{T} = \{\tilde{v}_1, \ldots, \tilde{v}_N\}$. Recall that a set of *lattice generators* in $\mathbb{R}^d$ is a set of d vectors in $\mathbb{R}^d$ such that any lattice node can be expressed uniquely as an integer combination of the lattice generators. We obtain a pair of lattice generators X_1, X_2 by using the well–known *Hermite Normal Form* algorithm (see e.g. [9]),

$$\mathbf{X} = \begin{bmatrix} X_1 & X_2 & 0 \end{bmatrix} = \begin{bmatrix} \tilde{v}_1 & \tilde{v}_2 & \ldots & \tilde{v}_N \end{bmatrix} \mathbf{U} = \widetilde{\mathbf{V}}\mathbf{U}$$

where $\mathbf{U}$ is a unimodular integral matrix. We refer in particular to the algorithm variant by [10]. In algebraic form, denoting by U_{ij} the element of U in the $i-j$ position, one has

$$X_j = \sum_{i=1}^{N} v_i U_{ij}.$$

Also let Δ be the determinant of the Hermite matrix X, i.e. the area of the mesh of the lattice generated by $\tilde{v}_1, \ldots, \tilde{v}_N$. From Minkowski's convex body theorem, an ϵ–neighborhood of an arbitrary point in the plane contains a lattice point if $\pi\epsilon^2 \geq 4\Delta$. Recall that Δ can be controlled by the accuracy of approximation τ_g.

Given the initial configuration $(0, 0, 0, F_1)$ and a final configuration $C_f = (x_f, y_f, \gamma_f, F_f)$, consider the algorithm (illustrated in fig. III):

Planning Algorithm.

1. If $\beta > 2\epsilon$ or $2\sqrt{\frac{\Delta}{\pi}} \geq \epsilon$, the Planning Problem is not solvable for a generic C_f (if the following steps are performed, a solution may or may not ensue).
2. Compute (x_1, y_1, γ_1) such that $L_f : (x_1, y_1, \gamma_1, F_1) \mapsto (x_f, y_f, \gamma_f, F_f)$. Observe that $L_f(0, 0, 0, F_1) = (x_f - x_1, y_f - y_1, \gamma - \gamma_1, F_f)$, which fact provides the equation for computing (x_1, y_1, γ_1).
3. Let $k = \arg\min_\kappa |\kappa\bar{\beta} - \gamma_1| \pmod{2\pi/\bar{\beta}}$. If $|k\bar{\beta} - \gamma_1| \pmod{\pi} > \epsilon$, the Planning Problem has no solution — exit; otherwise compute (x_2, y_2) such that $\bar{R}^k : (x_2, y_2, 0, F_1) \mapsto (x_1, y_1, \gamma_2, F_1)$, where $|\gamma_2 - \gamma_1| < \epsilon$. Observe that $\bar{R}^k : (0, 0, 0, F_1) \mapsto (x_1 - x_2, y_1 - y_2, \gamma_2, F_1)$ which fact provides the equation for computing (x_2, y_2).
4. Let $(k_1, k_2) = \arg\min_{\kappa_1, \kappa_2} \|\kappa_1 X_1 + \kappa_2 X_2 - (x_2, y_2)\|$. If $\|k_1 X_1 + k_2 X_2 - (x_2, y_2)\| > \epsilon$, the Planning Problem has no solution — exit; otherwise, apply the original generators of the lattice, $v_1, ..., v_N$, $U_i = U_{i1} k_1 + U_{i2} k_2$ times each. A point (x_3, y_3) is thus reached such that $\|(x_3, y_3) - (x_2, y_2)\| \leq \epsilon$.
5. Manipulate the object by applying (after purging all trivial substrings) the string of tumbles corresponding to $v_i^{U_i}$, $i = 1, ..., N$, $\bar{R}^k$, L_F, in the order.

This algorithm provides a complete solution to the Planning Problem for polyhedral parts whose geometric description is not affected by uncertainties ($\tau_a = 0$), and whose translation generators have not been approximated ($\tau_g = 0$).

For more general polyhedra, the algorithm is applied unchanged, but the computed solution may not satisfy the required accuracy ϵ. A test on the possibility of guaranteeing accuracy in the general case will be discussed shortly.

A. Accuracy of Solution

As already mentioned, by applying the proposed algorithm to general polyhedral parts with toleranced dimensions, it can not be guaranteed that their manipulation will actually lead to satisfy the planning problem. To answer this question, a detailed analysis of how measurement and approximation errors propagate and affect the configuration reached upon manipulation according to the above algorithm.

We start by explicitly calculating bound on errors on geometric parameters based on tolerances on input data for our problem, which are in our case the coordinates in space of the vertices of the polyhedron. Let τ_V denote the tolerance on the position of vertices V_i of the polyhedron,

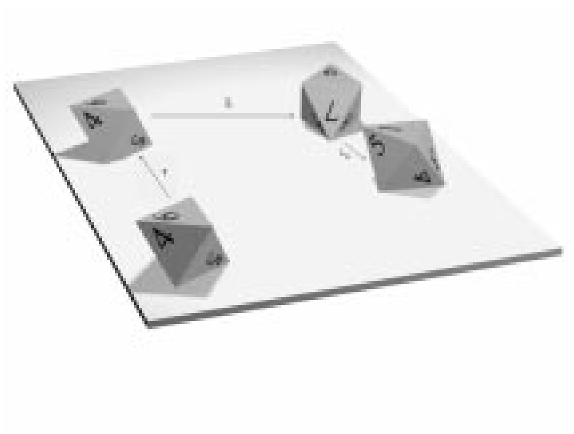


Fig. 5. Given the initial $((0,0,0,F_1))$ and the final $((x_f, y_f, \gamma_f, F_6))$ configurations the sequences of tumbles $T\bar{R}L_6$ are performed.

i.e.

$$\|\hat{V}_i - V_i\|_\infty \leq \tau_V, \quad i = 1,\dots,m.$$

Quantities to be computed that are affected by the error of measure are the length of edges D_{jk} incident at vertices V_j and V_k, and the defect angles β_i. A bound on the error on the edge length is clearly $\Delta_D = 2\tau_V$. The computation of defect angles may lead to errors bounded by $\Delta_\beta = \frac{2(l-1)}{d_{min}}\tau_V$, where $d_{min} = \min_{j,k} |D_{jk}|$, i.e. by the error on the angle between two edges incident at vertex V_i times the number of faces vertex V_i belongs to.

In the computation of $\bar{\beta}$, the GCD of defect angles, it is assumed that a rational representation $\frac{p_i}{q_i}\pi$ of defect angles β_i is taken such that

$$|\beta_i - \frac{p_i}{q_i}\pi| \leq \Delta_\beta, \; i = 1, ..., m-1$$

Based on

$$\bar{\beta} = \sum_{i=1}^{m-1} n_i \beta_i \pmod{2\pi},$$

we can write

$$|\bar{\beta} - \frac{p}{q}\pi| \leq \Delta_\beta \sum_{i=1}^{m-1} n_i \leq 2q(m-1)\Delta_\beta \pmod{2\pi}$$

In the manipulation sequence, rotations $\bar{R}$ appear at most $2q$ times in step 2 of the algorithm, and $U_i h_i$ times in step 3. While from definition in eq.1 one has directly $h_i \leq 2q$, a bound on U_i can be given by bounding the entries of matrix U and the projection of the final reached point on the basis X_1, X_2 for the lattice.

The Domich–Kannan–Trotter algorithm [10] brings the matrix of vectors $\tilde{v}_1, \dots, \tilde{v}_N$, in polynomial time $(2Nd_{max}^2)$, in the Hermite Normal Form and the entries of the final Hermite matrix $\mathbf{X}$ are bounded by d_{max}^{2N} with $d_{max} = \max_{j,k} |D_{jk}|$. By the algorithm the transforming unimodular matrix $\mathbf{U}$ such that $\mathbf{X} = \widetilde{\mathbf{V}}\mathbf{U}$ is easily computed and a bound on its entries is given by $d_{max}^{8N^2\Delta^2}$.

Let C_f the desired final configuration (with orientation invaried with respect to the initial one) at distance d_f from the origin. Let $\hat{C}_f = \tilde{k}_1 X_1 + \tilde{k}_2 X_2$ the point reached such that $\|\hat{C}_f - C_f\| \leq \sqrt{\frac{\Delta}{\pi}}$. Then, in matricial form, $(k_1 k_2)^T = (X_1 X_2)^{-1}\hat{C}_f$ and the infinity norm of the vector $(k_1 k_2)$, i.e. the biggest of the two coefficient representing the number of times the generators X_1, X_2 are employed, is bounded by the infinity norm of the matrix $(X_1 X_2)^{-1}$ times d_f. Then k_1, k_2 are bounded by $d_f \frac{d_{max}^{2N}}{\Delta}$.

At last a bound on U_i is given by

$$U_i \leq 2d_f \frac{d_{max}^{2N}}{\Delta} d_{max}^{8N^2\Delta^2} = 2d_f \frac{d_{max}^{2N(1+4N\Delta^2)}}{\Delta}. \tag{4}$$

Based on these computations, a bound on the error by which the orientation of the final configuration is reached can be provided as

$$\begin{aligned} \Delta\theta_f &\leq 2q\left(1 + 2d_f \frac{d_{max}^{2N(1+4N\Delta^2)}}{\Delta}\right)(2q(m-1)\Delta_\beta) = \\ &= 2q\left(1 + 2d_f \frac{d_{max}^{2N(1+4N\Delta^2)}}{\Delta}\right)\left(2q(m-1)\frac{2(l-1)}{d_{min}}\tau_V\right). \end{aligned} \tag{5}$$

It follows that reaching a final configuration at a distance d_f within some error ϵ in the orientation is possible only if $\Delta\theta_f \leq \epsilon$. Conversely, an estimate of the necessary accuracy in τ_V and Δ, i.e. τ_g, can be obtained for given ϵ and d_f.

Next the computation of the generators of translations is considered. From the theory reported above, we can express the generators as vectors of length equal to the distance between some fixed point $\bar{C}$ and the point C_i of the plane development of the polyhedron representing the vertex V_i. This length is affected by the error of the measure of the coordinates of the vertices magnified through computation of the plane development. We can bound this error by l (number of faces) times the error of measure τ_V. The direction of the generators is given by the line connecting the fixed point with the vertices on the plane development and then rotated of angles $k\beta + \frac{\beta_i}{2}$, for some $i = 1, \dots, m-1$, with $k = 1, .., q$ if p is even, or $k \leq 2q$ otherwise. Hence, the direction of the generators is affected by an error bounded by $\Delta_{gen} = \Delta_\beta[\frac{1}{2} + (2q)^2(m-1)]$.

The Hermite Normal Form for $\mathbf{V}$ is computed and the error in the computation and approximation to rationals of the coordinates of vectors v_i propagate through the algorithm.

In particular let $\hat{v}_i$ be the computed generator and $\delta\hat{v}_i$ be a vector such that the true generator of translation v_i can be written as $v_i = \hat{v}_i + \delta\hat{v}_i$ where $\|\delta\hat{v}_i\| \leq l\tau_V + d_{max}\Delta_{gen}$ In fact, $|\|\hat{v}_i\| - \|v_i\|| \leq l\tau_V$ and the absolute value of the angle between v_i and $\hat{v}_i$ is such that

$$|\angle(\hat{v}_i, v_i)| \leq \Delta_{gen}.$$

Let also $\tilde{v}_i$ the vector whose components are rational numbers approximating the coordinates of vector $\hat{v}_i$. Then for v_i it holds: $v_i = \hat{v}_i + \delta\hat{v}_i + \delta\tilde{v}_i$ with $\delta\tilde{v}_i$ the vector representing the approximation error. Observe that $\|\delta\tilde{v}_i\| \leq \sqrt{2}\tau_g$.

The Hermite Normal Form is computed on matrix $\tilde{\mathbf{V}} = \mathbf{V} - \delta\tilde{\mathbf{V}} - \delta\hat{\mathbf{V}}$ with $\delta\tilde{\mathbf{V}}$ and $\delta\hat{\mathbf{V}}$ are the matrices whose columns are respectively the vectors $\delta\tilde{v}_i$ and $\delta\hat{v}_i$. Then the basis for the lattice is computed up to an error given by $\left(\delta\tilde{\mathbf{V}} + \delta\hat{\mathbf{V}}\right)\mathbf{U}$. An upper bound on these errors is easily computed recalling the upper bounds on the euclidean norm of the columns of matrices $\delta\tilde{\mathbf{V}}$ and $\delta\hat{\mathbf{V}}$. The expression $E = N d_{max}^{8N^2\Delta^2}\left(l\tau_V + d_{max}\Delta_{gen} + \sqrt{2}\tau_g\right)$ is thus obtained.

It is clear that the error on the computation of X_1, X_2 is magnified when these two vectors are applied for steering the polyhedron in the desired configuration. Being then C_f the desired final configuration at distance d_f from the origin and $\hat{C}_f = \tilde{k}_1 X_1 + \tilde{k}_2 X_2$ the computed reachable point such $\|\hat{C}_f - C_f\| \leq \sqrt{\frac{\Delta}{\pi}}$, for the point $\bar{C}_f$ reached through actual manipulation it holds

$$\|\bar{C}_f - C_f\| \leq \|\bar{C}_f - \hat{C}_f\| + \|\hat{C}_f - C_f\| \leq \sqrt{\frac{\Delta}{\pi}} + 2Ed_f\frac{d_{max}^{2N}}{\Delta}$$

We obtain that a positioning precision of ϵ can be obtained if

$$\Delta_{C_f} = \sqrt{\frac{\Delta}{\pi}} + 2Ed_f\frac{d_{max}^{2N}}{\Delta} \leq \epsilon \tag{6}$$

Thus, based on (5) and (6), a final step of the proposed algorithm should be applied for polyhedra with uncertain geometric description as

Planning Algorithm (continued).

5. If $|k\bar{\beta} - \gamma_1| \pmod{\pi} + \Delta_{\theta_f} \leq \epsilon$ and $\|k_1X_1 + k_2X_2 - (x_2, y_2)\| + \Delta_{C_f} \leq \epsilon$, the algorithm succeeded in finding a solution to the Planning Problem.

If either checks in Step 5 of the algorithm fail, a different choice of some parameters in the algorithm may lead to success. In particular, reducing tolerances τ_V is obviously of great help, whenever possible. Another less trivial possibility is that of *reducing* the representation accuracy of the generators of translations, by increasing the lattice mesh area Δ. Although this decreases the lattice resolution, it also affects the number of times each generator is employed in manipulation, hence the accumulation of geometrical errors. The latter effect is dominant on the former for very small Δ (see (6)).

As a conclusion, we claim the following

Proposition 1: With respect to the Planning Problem as stated in this paper, the proposed Planning Algorithm is

- *complete* if the geometric description of the polyhedron is exact ($\tau_V = 0$) and if the polyhedron satisfies the discreteness condition of Remark 3;
- *resolution complete* if $\tau_V = 0$;
- not complete in the most general case (an optimization in the size of the lattice mesh might be necessary to conclude).

IV. Complexity of the Algorithm

In this section the analysis of the complexity of the algorithm previously described will be made. The total complexity is given by a time complexity of some parts of the algorithm and the number of tumbles (needed for finding a solution to the planning problem) times the time complexity of a single ET (which we denote for the moment with r). The total time complexity is comprised of two parts: one part which refer to the computation of the generators of rotations and translations and of the planning path for a given polyhedron and another one relative to the manipulation part. The complexity relative to the computation of the generators of translations and rotations is comprised of the computation of the polyhedron description and development, defect angles, approximation to rational of defect angles, $\bar{\beta}$, the set T, the lattice and the planning. For the second part we will consider the cost in terms of tumbles that are necessary for manipulating the part through the tumbles planned by the planning part of the algorithm.

A. Computational Part Complexity

For the computation of the geometrical parameters of the polyhedron the time running of the algorithm is valued in terms of number of edges, vertices and q which is the denominator of $\mathrm{GCD}(\beta_i/\pi)$ whose size is controled in order to obtain a certain accuracy in the orientation of the final configuration of the polyhedron. The computation of the lengths of the edges of the polyhderon is clearly linear with lk, i.e. the number of faces times the number of edges of polyhderon. In fact, for this computation stored data relative to the polyhedron parameters have to be explored: in particular, by the way datas are stored, the list of the faces has to be scanned and for each face the list of edges belonging to the faces has to be scanned. The polyhedron development is in turn linear with the number of faces l. The computation of defect angles also depends strictly on data storage. For each of the m vertices at most l faces have to be scanned and for each of the incident faces (which are in the worst case $l-1$) at most k edges have to be scanned in order to find the pair incident at some vertex. For all pairs of edges incident at the vertices and belonging to the same face the angle comprised has to be computed. The total complexity is given by that for the exploration which is at most mlk and that for the computation of the sum of the angles which is linear with ml then we have the comlexity for this part of $O(mlk)$.

Another important step is that of the approximation through the continued fraction expansion of defect angles. The number of iterations in the algorithm is such that the error of approximation is bounded by $\Delta_\beta = \frac{2(l-1)}{d_{min}}\tau_V$. As we have already observed this number is at most $\sqrt{\frac{1}{\Delta_\beta}}$.

For the computation of $\bar{\beta}$ and for the computation of the coefficients for the expression of $\bar{\beta}$ as in equation 2, the Euclidean Algorithm has to be repeated recursively $m-2$ times. Each time it is repeated its running time is at most $\log_{10}(pq)$, i.e. the number of digits of the smallest number whose GCD is computed. The total complexity is then of order $O(m\log_{10}(pq))$. The coefficients necessary for the computation of those of equation 2, are stored in the forward steps of the Euclidean Algorithm, and their running time computation is contained in that of the computation of $\bar{\beta}$.

Next $(m-1)q$ generators of the translations have to be computed. Recall that the remaining $(m-1)q$ generators are exactly the inverse of the firsts and their computation is trivial. For analysing the running time for the computation of each of these generators, recall that it is important the knowledge of the point $\bar{C}$ of the plane about which $\bar{R}$ rotates face F_1 of angle $\bar{\beta}$. Clearly the complexity of the computation for each of the generators is then given by the number of tumbles necessary for $\bar{R}$. We suppose for each tumble a time running cost as for the product of a 3–dimensional square matrix applied to the space coordinates of the m vertices in order to update the position on the plane of the polyhedron. Actually the matrix product is sufficient to be applied to only 3 of the m vertices and update all the others, in terms of relative position with respect to some fixed reference system, when the final tumble has been simulated. Then the simulation of a single tumble has constant complexity.

The number of tumbles $\bar{N}$ that $\bar{R}$ requires can be estimated by its expression in terms of product of rotations R_j's: $\bar{R} = \prod_{j=1}^{m-1} R_j^{n_j}$ where n_j are the integers such that $\sum_{j=1}^{m-1} n_j\beta_j = \frac{p}{q}\pi$. We have then $\bar{N} = \sum_{j=1}^{m-1} n_j N_j$ where N_j is the number of tumbles needed for the execution of a single rotation R_j. In turn N_j is given by twice the number of tumbles needed for the transit path L_{j_1} (where F_{j_1} is one of the face to which V_j belongs to) plus those needed for rotating about all the edges incident at vertex V_j. Then $N_j \leq 2(l-1) + (l-1) = 3(l-1)$.

It follows immediately $\bar{N} = \sum_{j=1}^{m-1} n_j N_j \leq (m-1)(2q)3(l-1)$ and $\bar{N} = O(qlm)$. At last for the whole set of generators, we have a computational cost of

$$rq(m-1)\left((m-1)(2q)3(l-1)\right),$$

i.e. of order $O(q^2m^2l)$.

Next step is the lattice generators computation which is done by bringing the matrix of translation in Hermite Normal Form which requires $2Nd_{max}^2$ steps in each of which only elementary matrix operation are done. Then the complexity of the algorithm can be supposed of order $O(2Nd_{max}^2)$.

Next step in the computation of the complexity of the algorithm concerns the computation of a solution. Recalling the steps of the planning algorithm it is necessary to implement a path L_f for step 1., and a path $\bar{R}^k$ for some $k \leq 2q$ for step 2. Then this part of the algorithm has a running time cost bounded by $r\left((2q)\bar{N} + (l-1)\right) \leq 9(l-1)\left((2q)^2(m-1)3+1\right)$ Then this part of the algorithm has complexity of order $O(q^2ml)$.

In this part of the algorithm integers k_1, k_2 such that

$$(\bar{x}_f, \bar{y}_f) = (X_1\ X_2)\begin{pmatrix} k_1 \\ k_2 \end{pmatrix} \tag{7}$$

is verified with $\|(\bar{x}_f, \bar{y}_f) - (x_f, y_f)\| \leq \sqrt{\frac{\Delta}{\pi}}$, the nearest point from the final desired configuration is $(\bar{x}_f, \bar{y}_f)$ and is reached in $k_1 + k_2 \leq 2Max(k_1, k_2)$ times the generators X_1 and X_2. Recalling the bound found in the previous section $Max(\tilde{k}_1, \tilde{k}_2) \leq d_f \frac{d_{max}^{2N}}{\Delta}$.

B. Manipulation Complexity

The number of tumbles planned by the algorithm are performed. In order to provide their total number we recall the following bounds we have found above:
i Number of tumbles for L_f is at most $l-1$
ii Number of tumbles for $\bar{R}^k$ is at most $(2q)^2(m-1)3(l-1)$
iii Number of tumbles for $t = k_1X_1 + k_2X_2$ with $(X_1X_20) = \mathbf{VU}$. Then

$$t = \mathbf{VU}\begin{bmatrix} k_1 \\ k_2 \\ 0 \end{bmatrix}.$$

Each vector v_i is of type $\bar{R}^s\left(\bar{R}^{h_j}R_j^{-1}\right)\bar{R}^{-s}$. A bound on the tumbles that each vector of translation requires is given by $N_i \leq (2q)(m-1)3(l-1)(2s+h_j) + 3(l-1) \leq 3(l-1)(1 + 2(m-1)(2q)^2)$. We can briefly say that each generator of translation is of $O(q^2ml)$. The number $U_i = k_1U_{i1} + k_2U_{i2}$ of time each of them is applied has been already estimated in the previous section (see equation 4). Then we have a total number of tumbles for the translational part which is given by $\sum_{i=1}^{N} N_iU_i = O\left(q^2mld_f \frac{d_{max}^{N^2\Delta^2}}{\Delta}\right)$.

V. Conclusions

We studied the problem of displacing and reorienting a polyhedral part rolling on a plane to the purposes of robotic manipulation. One contribution of this paper is to provide a correct statement of this problem, that takes into account the mathematical structure of the system as well as practical considerations on finite–acuuracy measurements of bodies. We also proposed an algorithm, and discussed its properties in relation to its completeness and computational complexity.

The manipulation sequences generated by the planner are in general rather long, and by no means optimal in the sense of minimizing the number of tumbles. Further work in that direction is envisioned in the next future.

Acknowledgments

The Authors wish to thank Dr. Yacine Chitour for helpful discussions and suggestions.

References

[1] A. Bicchi and K. Goldberg, eds., Minimalism in Robotic Manipulation, *Proceedings of the IEEE 1996 Workshop on Minimalism in Robotic Manipulation, Minneapolis, MA, April 22–28, 1996.*

[2] A. Bicchi, and R. Sorrentino, Dexterous Manipulation through Rolling, in *Proc. IEEE Int. Conf. on Robotics and Automation,* 1995.

[3] A. Bicchi, D. Prattichizzo, and S. S. Sastry, Planning Motions of Rolling Surfaces, in *IEEE Conf. on Decision and Control*, 1995.

[4] N. Sawasaki M. Inaba and H. Inoue, Tumbling Objects using a Multifingered Robot. in *Proc. 20th ISIR,* 1989.

[5] Y. Aiyama, M. Inaba and H. Inoue, Pivoting: A New Method of Graspless Manipulation of Object by Robot Fingers, in *Proc. IEEE/RSJ Int. Conf. on Int. Robots and Systems, IROS'93,* 1993.

[6] M.A., Erdmann, M.T. Mason and G. Vanvevcek, Mechanical Parts Orienting: the Case of a Polyhedron on a Table. *Algorithmica* volume 10, number 2, 1993.

[7] Y.Chitour, A.Marigo, D.Prattichizzo, A.Bicchi, Rolling Polyhedra on a Plane, Analysis of the Reachable Set, in *Workshop on Algorithmic Foundations of Robotics* '96

[8] A.Marigo, Y.Chitour, A.Bicchi. Manipulation of Polyhedral Parts by Rolling, presented *International Conference of Robotics and Automation* '97

[9] A. Schrijver, *Theory of Linear and Integer Programming* Wiley Interscience Publ., 1986.

[10] P.D. Domich, R. Kannan, and Jr. L.E. Trotter, Hermite Normal Form Computation Using Modulo Determinant Arithmetic, *CORE Discussion paper 8507, Center for Operations Research and Econometrics, Louvain–la–Neuve (Belgium)*, 1985.

[11] H.S.M.Coxeter, *Regular Polytopes*, Macmillan Mathematics Paperbacks

[12] V.V.Nikulin, I.R.Shafarevich, *Geometries and Groups* Berlin: Springer '87

[13] K. Goldberg, Feeding and Sorting Algorithms for the parallel–jaw Gripper, in *Proc. 6th Int. Symposium on Robotics Research,* 1993.

[14] S.C. Jacobsen, J. Wood, K. Bigger, and E. Iverson, The Utah/MIT hand: Work in progress, *International Journal of Robotic Research,* 4(3):21-50, 1984.

[15] Z. Li and J. Canny, Motion of Two Rigid Bodies with Rolling Constraint, *IEEE Trans. on Robotics and Automation,* volume 6, number 1, 1990.

[16] A. Marigo, Rolling Polyhedra on a Plane, *Internal Report, Centro "E. Piaggio", University of Pisa*, 1996.

[17] D.J. Montana, The Kinematics of Contact and Grasp, *Int. J. of Robotics Research,* 7(3), pp. 17:32, 1988.

[18] G.L. Nemhauser, and L.A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley and Sons, 1988.

[19] M.A. Peshkin and A.C.Sanderson, Planning Robotic Manipulation Strategies for Workpieces that Slide, *IEEE Journal of Robotics and Automation,* volume 4, number 5, 1988.

[20] P. Tournassoud, T. Lozano–Perez and E. Mazer, Regrasping, in *Proc. IEEE Int. Conf. on Robotics and Automation,* 1987.