

Geometric design of motions constrained by a contacting surface pair

Michael Hofer ^{a,b,*}, Helmut Pottmann ^b, Bahram Ravani ^a

^a*Department of Mechanical and Aeronautical Engineering,
University of California, Davis, CA 95616, USA*

^b*Institute of Geometry, Vienna University of Technology,
Wiedner Hauptstrasse 8-10/113, A-1040 Wien, Austria*

To the memory of Josef Hoschek

Abstract

We discuss the following problem which arises in robot motion planning, NC machining and computer animation: Given are a fixed surface Ψ and N positions Φ_i of a moving surface Φ such that the Φ_i are in point contact with Ψ . Compute a smooth and fair Euclidean gliding motion $\Phi(t)$ of the surface Φ on the surface Ψ which interpolates (or approximates) the given positions Φ_i at time instances t_i . First we generalize interpolatory variational subdivision algorithms for curves to curves on surfaces. Second we study an unconstrained motion design algorithm which we then extend to the main contribution of this paper, an algorithm for the design of a motion constrained by a contacting surface pair. Both motion design algorithms use a feature point representation of the moving surface, subdivision algorithms for curves, instantaneous kinematics, and ideas from line geometry. Geometric methods are used for the numerical solution of the arising optimization problems.

Key words: Motion design, Variational subdivision, Gliding motion, Kinematics, Line Geometry

* Corresponding author: *phone* +43 1 58801 11315, *fax* +43 1 58801 11399
Email address: hofer@geometrie.tuwien.ac.at (Michael Hofer).
URL: <http://www.geometrie.tuwien.ac.at/hofer> (Michael Hofer).

1 Introduction

Motion design is an important and widely studied problem, e.g., in robotics, manufacturing geometry, and computer animation. We are interested in the design of a *smooth* and *fair gliding* motion of one surface on another surface such that given positions already in contact are either interpolated or approximated. We confine ourselves to differentiable one-parameter motions in Euclidean 3-space and to parametric surfaces. Using variational subdivision, instantaneous kinematics, and the relation to line geometry, we show how to construct such motions. Gliding motion design has applications e.g. in the NC machining of complex shapes (cf. Marciniak (1991)), and in medical applications where the gliding motion of joints is studied (cf. Hoschek and Weber (1987)).

1.1 Review of previous work

Motion design is an important research area in fields such as Computational Geometry (see e.g. Halperin et al. (1997); Sharir (1997)), Robotics and Kinematics (see e.g. Latombe (2001); Park and Ravani (1995)), Computer Graphics (see e.g. Shoemake (1985); Marchand and Courty (2002)) and Computer Aided Geometric Design (see e.g. Röschel (1998); Jüttler and Wagner (2002)). For applications it is important to be able to control the smoothness level of a motion. Forces and vibrations depend on second derivatives and jumps in the velocity should be avoided with NC machining and in computer animation. Thus one goal is the design of C^2 motions. Smooth motions that make use of NURBS techniques have been investigated by e.g. Jüttler and Wagner (1996). To describe rigid body motions it is necessary to use *rational* rather than polynomial representations. However, rational representations are much less suitable for variational design and efficient optimization techniques than polynomial ones. Therefore, Hyun et al. (2001) recently investigated the construction of affine spline motions with minimal distortion. Motion design based on the quaternion representation of the spherical component and non-linear extensions of spline constructions in affine spaces to the sphere (see the references in Röschel (1998)), are also difficult to deal with for optimization purposes. Algorithms for motion fairing using the quaternion representation have been proposed by Fang et al. (1998) and Hsieh and Chang (2003), where the latter are using genetic algorithms. Belta and Kumar (2002) recently presented a singular value decomposition projection method for interpolation on the group of Euclidean motions $SE(3)$.

1.2 Contributions of the present paper

Hofer et al. (2002) have outlined two new approaches to motion design which deal with *variational motion design* and that allow to incorporate *constraints* into the motion planning process. The first approach has been studied in Hofer et al. (2003), the second approach will be studied here and is extended to handle motions constrained by a contacting surface pair which is the main contribution of this paper. The *singularities* of motions constrained by a contacting surface pair have been studied by Pottmann and Ravani (2000). Here we present an algorithm for the *design* of a gliding motion for the entire motion cycle rather than around singularities. All other previous works have not considered gliding motions where a rigid moving body maintains contact with another body during its motion cycle. Wallner (2002) has investigated gliding spline motions using an active motion approach. However, the resulting motions are in general only near-Euclidean near-contact spline motions. Our work in terms of the formulation of the fairness is also different from previous approaches on un-constraint motion design in that our formulation of ‘fairness’ is based on fair trajectories of chosen feature points of the moving body. A rigid body motion is a curve in the Euclidean motion group and variational motion design could be based on expressing the fairness of that curve (see Park and Ravani (1997)). However, for the applications we have in mind, not the motion as such, but its action on a certain rigid body is employed in fairness criteria. Another contribution of this paper is the extension of known *interpolatory* variational subdivision algorithms for curves to special cases of variational curve design via subdivision on k -dimensional surfaces in \mathbb{R}^d .

The paper is organized as follows. In Sect. 2 we review some aspects of existing interpolatory variational subdivision algorithms for curves and extend them to curves on surfaces. In Sect. 3 we summarize relations between spatial kinematics and line geometry that we use for our motion design algorithm. In Sect. 4 we present a subdivision algorithm for *unconstraint* motion design, which in Sect. 5 is extended to a subdivision algorithm for the design of a *gliding motion* of one surface on another surface such that given positions in contact are interpolated or approximated. We conclude the paper in Sect. 6 with a summary of our work and an outlook towards future research.

2 Variational subdivision for curves

Kobbelt (1996) has introduced a class of *interpolatory* variational subdivision schemes. Thereby, new points for the refinement of a given polygon are determined by solving an optimization problem that minimizes quadratic energy functionals. This is a familiar approach from non-discrete curve design where

energy functionals are minimized for curve *fairing*. Kobbelt and Schröder (1998) have extended interpolatory variational subdivision from the *uniform* to the *non-uniform* parameter setting and discussed it in a multiresolution framework. These schemes are *global*, i.e., every new point depends on all points to be refined. Interpolation is guaranteed since the old points belong to the newly calculated finer version. The computational complexity is linear in the number of points of the polygon.

Important examples of variational subdivision are found by minimizing differences. In the uniform (non-uniform) case we minimize forward (divided) differences, see Kobbelt and Schröder (1998). Explicit formulae for the minimization of *second forward differences* in the *open* and *closed* curve case can be found in Hofer et al. (2003), who have also generalized *interpolatory* schemes to *approximating* ones. Kobbelt (1996) has shown that by minimizing second differences the limit curves are at least C^2 curves.

2.1 Variational subdivision for surface curves

Given are N points $\mathbf{f}_1, \dots, \mathbf{f}_N$ on a smooth two-dimensional parametric surface $\Phi \subset \mathbb{R}^3$. If $\Phi(u, v)$ is a parametrization of Φ , then $\mathbf{f}_i = \Phi(u_i, v_i)$. To compute a variational surface curve on Φ that interpolates the given points we propose the following algorithm (which we call ‘tangential method’, see Fig. 1).

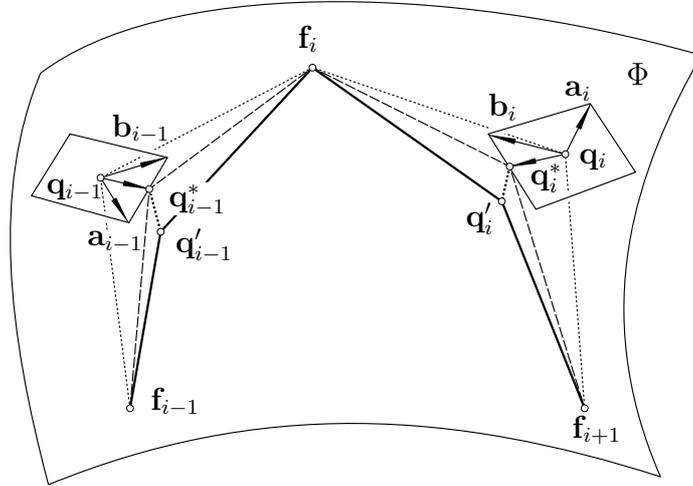


Fig. 1. Variational subdivision for surface curves.

First insert by some heuristic between the given points new points $\mathbf{q}_1, \dots, \mathbf{q}_{N-1}$ on Φ , i.e., $\mathbf{q}_j = \Phi(u_j, v_j)$. Then compute a basis $\mathbf{a}_i, \mathbf{b}_i$ of Φ 's tangent space at \mathbf{q}_i and define (yet unknown) displacement vectors

$$\mathbf{v}_i := \lambda_i \mathbf{a}_i + \mu_i \mathbf{b}_i. \tag{1}$$

where $i = 2, \dots, N - 2$. The solution of (3) are $\lambda_i, \mu_i, i = 1, \dots, N - 1$ which are then used to compute the points $\mathbf{q}_i^* := \mathbf{q}_i + \lambda_i \mathbf{a}_i + \mu_i \mathbf{b}_i$. Note that \mathbf{q}_i^* is only in the tangent space of Φ at \mathbf{q}_i . Hence we orthogonally project \mathbf{q}_i^* onto a point \mathbf{q}'_i of Φ , i.e., we compute \mathbf{q}'_i such that the following squared distance d^2 is minimized,

$$d^2(\mathbf{q}_i^*, \mathbf{q}'_i) = \min_{\tilde{\mathbf{q}}_i \in \Phi} d^2(\mathbf{q}_i^*, \tilde{\mathbf{q}}_i). \quad (7)$$

The minimization of (7) is a nonlinear problem and can be solved with a Newton type method, cf. Fletcher (2000). If \mathbf{q}_i^* is not on the medial axis of Φ then we find a unique point $\mathbf{q}'_i \in \Phi$. We repeat the minimization of (2) until the value of the objective function F_2 falls below a certain threshold. Then we insert the final positions \mathbf{q}'_i into the sequence of given points \mathbf{f}_i and continue with the next subdivision step.

Remark 1 *One could perform variational subdivision in the parameter domain and then map the such obtained curve to the parametric surface. We call this the ‘parameter domain method’. However, since the evaluation of a parametric surface is not an affine map, the such obtained surface curve does not minimize the variational energy functional any more, see Fig. 2. A summary of the ‘tangential method’ is given in Alg. 1.*

Remark 2 *All numerical experiments for this paper have been performed with a Matlab implementation on a 1.8 GHZ Pentium 4 with 1GB RAM. Timing results for our method (2) are given in Table 1. For a fast orthogonal projection of a point onto a surface (7) we refer the reader to the literature (cf. Johnson and Cohen (1998) and the references therein). Since we have just implemented a simple un-optimized algorithm for the solution of (7) we do not give timing results for this part of the algorithm. Table 1 contains the following information, which we explain at hand of the Moebius strip example: given 7 input points on Φ we apply 4 subdivision steps which results in 97 output points on Φ . In subdivision step 1, 2, 3, 4 the optimization (2) has been performed 4, 3, 2, 2 times. The total number of 11 calls of (2) took a computation time of 0.07s.*

| | Moebius strip | Torus | Wave | Ellipsoid |
|------------------------|---------------|---------|-------------|-----------|
| input/output points | 7/97 | 20/305 | 12/353 | 7/97 |
| subdivision steps | 4 | 4 | 5 | 4 |
| optimization steps (2) | 4,3,2,2 | 8,4,3,2 | 13,13,3,2,2 | 6,6,5,4 |
| total calls/time (2) | 11/0.07s | 17/0.2s | 33/0.24s | 21/0.11s |

Table 1
Numerical results for surface curves shown in Fig. 2.

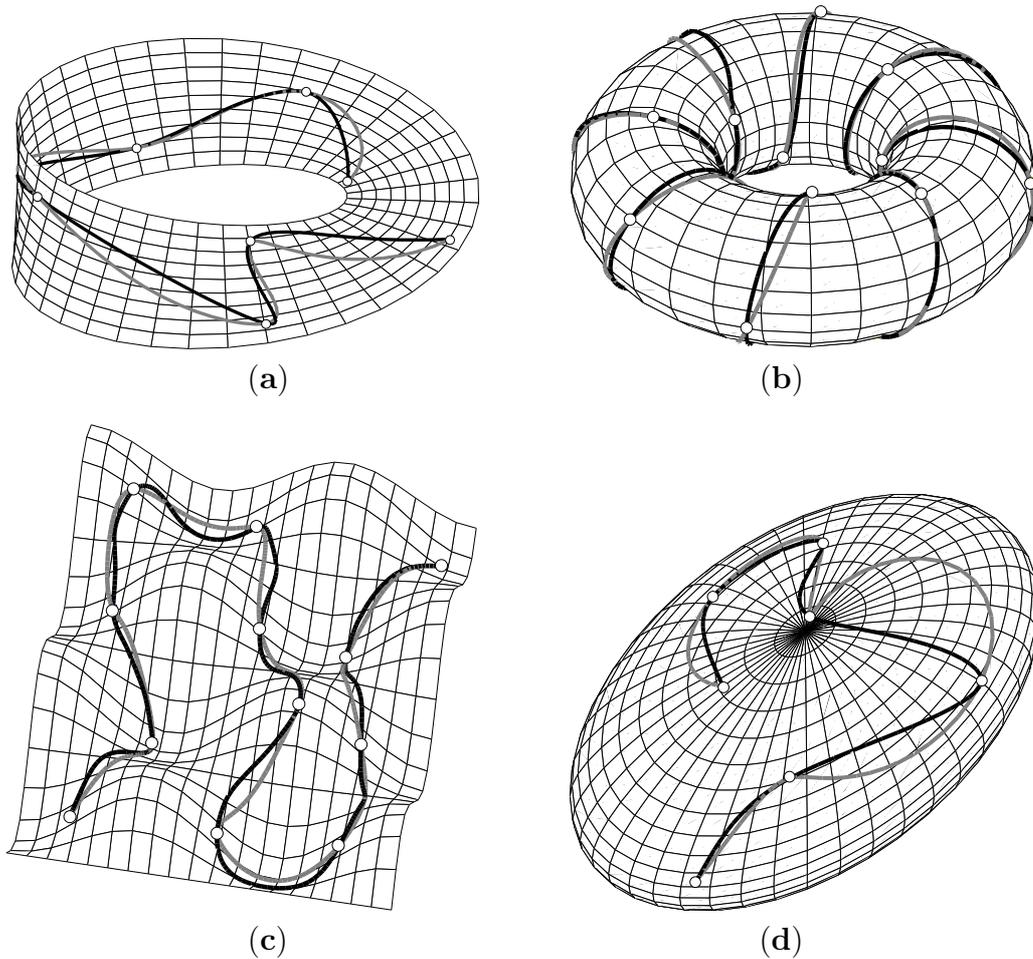


Fig. 2. Variational subdivision surface curves interpolating given surface points: ‘tangential method’ (black curve), ‘parameter domain method’ (gray curve): (a) Moebius strip, (b) torus, (c) wave surface patch, (d) ellipsoid.

3 The relation between spatial kinematics and line geometry

3.1 Instantaneous kinematics

Consider a differentiable one-parameter rigid body motion in Euclidean 3-space. Introducing Cartesian coordinate frames Σ in the moving system and Σ_0 in the fixed system, the time dependent position $\mathbf{x}_0(t)$ of a point $\mathbf{x} \in \Sigma$ in the fixed system Σ_0 is given by

$$\mathbf{x}_0(t) = \mathbf{R}(t)\mathbf{x} + \mathbf{r}(t). \quad (8)$$

Input: N_0 points $\mathbf{f}_1^{(0)}, \dots, \mathbf{f}_{N_0}^{(0)}$ on a surface Φ , number of subdivision steps S , maximum number of optimization steps M , threshold T for optimization termination

for $l = 1$ **to** S

(1) insert initial points $\mathbf{q}_i^{(l-1)}$ between $\mathbf{f}_i^{(l-1)}$ and $\mathbf{f}_{i+1}^{(l-1)}$ for $i = 1, \dots, N_{l-1} - 1$

(2) **while** $j < M$ **and** $val > T$ **do**

• compute $\mathbf{q}_i^{*(l-1)}$ by minimizing F_2 given by (2)

• compute $val = |F_2(\mathbf{q}_i^{(l-1)}) - F_2(\mathbf{q}_i^{*(l-1)})|$

• **if** $val > T$

· orthogonally project $\mathbf{q}_i^{*(l-1)} \notin \Phi$ onto $\mathbf{q}_i'^{(l-1)} \in \Phi$ via (7)

· $\mathbf{q}_i^{(l-1)} = \mathbf{q}_i'^{(l-1)}$

· $j = j + 1$

endif

end

(3) $\mathbf{f}_{2i-1}^{(l)} := \mathbf{f}_i^{(l-1)}, i = 1, \dots, N_{l-1}$

$\mathbf{f}_{2i}^{(l)} := \mathbf{q}_i^{(l-1)}, i = 1, \dots, N_{l-1} - 1$

end

Output: $N_S := (N_0 - 1)2^S + 1$ points $\mathbf{f}_1^{(S)}, \dots, \mathbf{f}_{N_S}^{(S)}$ on the surface Φ

Alg. 1: Variational subdivision algorithm for surface curves.

Here, the time dependent orthogonal matrix $\mathbf{R}(t)$ represents the spherical component of the motion, and $\mathbf{r}(t)$ describes the trajectory of the origin of the moving system. All arising functions shall be C^1 . By differentiation we get the velocity vectors. It is well-known that the velocity vector field is linear at any time instant. More precisely, at any time instant there exist vectors $\mathbf{c}, \bar{\mathbf{c}}$ such that the velocity vector $\mathbf{v}(\mathbf{x})$ of any point \mathbf{x} of the moving body can be computed as

$$\mathbf{v}(\mathbf{x}) = \bar{\mathbf{c}} + \mathbf{c} \times \mathbf{x}. \quad (9)$$

Note that in this formula all arising vectors are represented in the same system; this may be the moving or the fixed system. The meaning of $\mathbf{c}, \bar{\mathbf{c}}$ is as follows: $\bar{\mathbf{c}}$ represents the velocity vector of the origin, and \mathbf{c} is the so-called Darboux vector. Only very special one-parameter motions have a constant, i.e., time-independent velocity vector field. These motions are

- A *translation* with constant velocity (if $\mathbf{c} = \mathbf{0}$),
- A uniform *rotation* about an axis (if $\mathbf{c} \cdot \bar{\mathbf{c}} = 0$),
- A uniform *helical motion* (if $\mathbf{c} \cdot \bar{\mathbf{c}} \neq 0$).

The most general case is that of a uniform helical motion, which is the superposition of a rotation with constant angular velocity about an axis A and a translation with constant velocity parallel to A . If the moving body rotates about an angle α , the translation distance is $p\alpha$. The constant factor p is referred to as *pitch* of the helical motion. The Plücker coordinates $(\mathbf{a}, \bar{\mathbf{a}})$ of the axis A , the pitch p of the helical motion and the angular velocity ω are computed from $(\mathbf{c}, \bar{\mathbf{c}})$ by

$$\mathbf{a} = \frac{\mathbf{c}}{\|\mathbf{c}\|}, \quad \bar{\mathbf{a}} = \frac{\bar{\mathbf{c}} - p\mathbf{c}}{\|\mathbf{c}\|}, \quad p = \frac{\mathbf{c} \cdot \bar{\mathbf{c}}}{\mathbf{c}^2}, \quad \omega = \|\mathbf{c}\|. \quad (10)$$

For more details on helical motions and the close relation to line geometry we refer to Pottmann and Wallner (2001).

3.2 Line geometry

In three-dimensional Euclidean space E^3 with a Cartesian coordinate system, a straight line L can be represented by a *normalized direction vector* \mathbf{l} , $\|\mathbf{l}\| = 1$, and its *moment vector* $\bar{\mathbf{l}} := \mathbf{x} \times \mathbf{l}$ with respect to the origin. Thereby, \mathbf{x} denotes the coordinate vector of an arbitrary point on L . Note that $\bar{\mathbf{l}}$ is independent of the choice of \mathbf{x} . The 6 coordinates of L are its so-called *normalized Plücker coordinates*. They satisfy the Plücker relation $\mathbf{l} \cdot \bar{\mathbf{l}} = 0$. Any 6-tuple $(\mathbf{l}, \bar{\mathbf{l}}) \in \mathbb{R}^6$ with $\|\mathbf{l}\| = 1$ and $\mathbf{l} \cdot \bar{\mathbf{l}} = 0$ represents a line in E^3 , where $(\mathbf{l}, \bar{\mathbf{l}})$ and $(-\mathbf{l}, -\bar{\mathbf{l}})$ describe the same line.

One of the fundamental relations between spatial one-parameter motions and line geometry is, that for every time instant t , the path normals $(\mathbf{l}, \bar{\mathbf{l}})$, i.e., the lines through a point \mathbf{x} normal to $\mathbf{v}(\mathbf{x})$, of such a motion lie in a linear line complex \mathcal{C} . Such a linear complex is defined as the solution set of a linear homogeneous equation in Plücker coordinates,

$$0 = \mathbf{v} \cdot \mathbf{l} = (\bar{\mathbf{c}} + \mathbf{c} \times \mathbf{x}) \cdot \mathbf{l} = \bar{\mathbf{c}} \cdot \mathbf{l} + \mathbf{c} \cdot \bar{\mathbf{l}}. \quad (11)$$

The linear line complex is singular if either $p = 0$ or $p = \infty$. We speak of the pitch p and the axis A of a linear complex which can be computed via (10), since \mathcal{C} is defined by $(\mathbf{c}, \bar{\mathbf{c}})$. Conversely, any linear line complex \mathcal{C} , defined as the solution set of a linear homogeneous equation in Plücker coordinates

$$\bar{\mathbf{c}} \cdot \mathbf{l} + \mathbf{c} \cdot \bar{\mathbf{l}} = 0, \quad (12)$$

can be obtained as a path normal complex of a spatial one-parameter motion.

4 Variational subdivision for unconstraint motion design

Hofer et al. (2002) have outlined two variational subdivision algorithms for motion design. The first algorithm has been investigated in Hofer et al. (2003), the second one is studied in this section. This algorithm computes to given positions Φ_1, \dots, Φ_N of a moving body Φ a smooth and fair motion $\Phi(t)$ that interpolates or approximates Φ_i at time instances t_i . Current motion design algorithms consider fairness criteria which possess full kinematic invariance. This means that the result does not change under the application of Euclidean displacements in the moving or fixed system, respectively. As a consequence of that, one cannot impose fairness criteria based only on the trajectories of points on the moved object. If one works in Euclidean three-space, the advantage is, that fairness criteria can be employed much easier than if we would work on the six-dimensional manifold of Euclidean motions. Therefore we choose K feature points $\mathbf{f}^1, \dots, \mathbf{f}^K$ on the moving body Φ that represent it sufficiently well, and use them to employ fairness criteria in the motion design process. Fig. 3 shows a simplified robot gripper arm where the 24 vertices of the polyhedral representation are used as feature points. For the choice of feature points we refer the reader to Sect. 4.4. Another advantage of our ap-

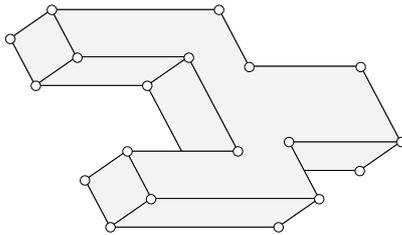


Fig. 3. Simplified robot gripper arm with 24 feature points.

proach is, that it can be used with the whole class of interpolatory variational subdivision algorithms introduced by Kobbelt (1996), and with the extension to approximating schemes as discussed in Sect. 2. We present the algorithm in terms of the minimization of second forward differences, since in that case the formulae are shorter. The generalization to the non-uniform parameter setting and to an approximating algorithm is straightforward, although a bit more tedious.

4.1 Outline of the algorithm

Variational subdivision for curves combines the insertion and position optimization (i.e., minimization of some quadratic energy functional) of new points in every subdivision step. In the motion design case we have to split this into two separate steps, insertion and optimization, since we do not just

have a single point but a rigid body. These two steps are repeated until the required density of positions is reached:

- (1) **Position insertion:** Insert between each two adjacent given copies one Euclidean copy $\Phi_{i|i+1}$ of the moving body. For given N positions this gives $N - 1$ initial intermediate positions.
- (2) **Position optimization:** Use $\Phi_{i|i+1}$ as starting values for an iterative optimization algorithm that uses linearized motions and simultaneous variational subdivision on all K feature points. Terminate the iteration if the objective function reaches a minimum, or alternatively, if the sum of squared distances between the feature points in the current and in the following iteration step falls below a certain threshold. Then add the final positions to the sequence of given positions, update the number of given positions to $2N - 1$ and continue with step 1.

Remark 3 *By a position of a rigid body we mean the location in space determined by six parameters, three for translation and three for rotation. In the literature, a position sometimes is only defined by the three translation parameters, and the three rotation parameters are referred to as orientation of the rigid body.*

4.2 Position insertion

Between each two adjacent given copies Φ_i and Φ_{i+1} one new copy $\Phi_{i|i+1}$ of the moving body Φ is inserted (in the curve case one new point would be inserted between each two given points). We get $\Phi_{i|i+1}$ by applying the Euclidean rigid body transformation m to Φ_i ,

$$\Phi_{i|i+1} := m(\Phi_i) := \mathbf{R}_i(\Phi_i) + \mathbf{t}_i, \quad (13)$$

where m is chosen such that $\Phi_{i|i+1}$ ‘is between’ Φ_i and Φ_{i+1} . The simplest choice of m is the identity transformation $\mathbf{R}_i = \mathbf{I}$ and $\mathbf{t}_i = \mathbf{0}$. However, the optimization step of the algorithm (described in Sect. 4.3) needs less iteration steps, if we compute ‘reasonable’ \mathbf{t}_i and \mathbf{R}_i . This computation of the translational and the rotational part is done separately. Consider the given positions \mathbf{s}_i of the *barycenter* \mathbf{s} of the moving body Φ and apply one iteration step of interpolatory variational subdivision for curves to find intermediate positions $\mathbf{s}_{i|i+1}$. Then the obvious choice for the translational part \mathbf{t}_i of the Euclidean transformation m is,

$$\mathbf{t}_i := \mathbf{s}_{i|i+1} - \mathbf{s}_i. \quad (14)$$

Now we discuss the computation of \mathbf{R}_i . Translate Φ_{i+1} by $\mathbf{t}_{i+1}^* := \mathbf{s}_i - \mathbf{s}_{i+1}$ to a position Φ_{i+1}^* such that $\mathbf{s}_{i+1}^* = \mathbf{s}_i$. Since Φ_i and Φ_{i+1} are Euclidean copies of Φ there is a unique rotation – with angle of rotation ρ_i and direction vector \mathbf{d}_i of the rotation axis – that rotates Φ_i into Φ_{i+1}^* . In our algorithm we rotate through an angle of $\rho_i/2$ about \mathbf{d}_i , which corresponds to the unit quaternion

$$(a_0, a_1, a_2, a_3) = \left(\cos \frac{\rho_i}{4}, \sin \frac{\rho_i}{4} \mathbf{d}_i \right). \quad (15)$$

The rotation matrix corresponding to (15) is

$$\mathbf{R}_i = \begin{pmatrix} a_0^2 + a_1^2 - a_2^2 - a_3^2 & 2(a_1a_2 + a_0a_3) & 2(a_1a_3 - a_0a_2) \\ 2(a_1a_2 - a_0a_3) & a_0^2 - a_1^2 + a_2^2 - a_3^2 & 2(a_2a_3 + a_0a_1) \\ 2(a_1a_3 + a_0a_2) & 2(a_2a_3 - a_0a_1) & a_0^2 - a_1^2 - a_2^2 + a_3^2 \end{pmatrix}. \quad (16)$$

Thus we find the starting positions $\Phi_{i|i+1}$ in the following way: For $i = 1, \dots, N - 1$ first compute the translational part \mathbf{t}_i by (14), then the rotational part \mathbf{R}_i by (15) and (16), and finally apply the Euclidean transformation (13) to get the starting position $\Phi_{i|i+1}$. For easier reading of the following computations we denote the feature points in the positions $\Phi_{i|i+1}$ by \mathbf{q}_i^k for $i = 1, \dots, N - 1$ and $k = 1, \dots, K$.

4.3 Position optimization

This step in our motion design algorithm simultaneously optimizes the positions of the inserted copies $\Phi_{i|i+1}$. The iterative algorithm uses instantaneous kinematics and simultaneous interpolatory variational subdivision on the K sequences of feature points $\mathbf{f}_1^k, \mathbf{q}_1^k, \mathbf{f}_2^k, \dots, \mathbf{f}_{N-1}^k, \mathbf{q}_{N-1}^k, \mathbf{f}_N^k$ ($k = 1, \dots, K$). Positions $\mathbf{f}_1^k, \mathbf{q}_1^k, \mathbf{f}_2^k \dots$ of the same feature point \mathbf{f}^k at different time instances are called *homologous points*. If we assume uniformly distributed time instances t_i of our input positions, then we generalize uniform interpolatory variational subdivision for curves. To the feature points \mathbf{q}_i^k , $k = 1, \dots, K$, of each single copy $\Phi_{i|i+1}$, $i = 1, \dots, N - 1$ we attach (see Fig. 4) vectors

$$\mathbf{v}_i^k := \mathbf{v}(\mathbf{q}_i^k) = \bar{\mathbf{c}}_i + \mathbf{c}_i \times \mathbf{q}_i^k, \quad (17)$$

belonging to the linear velocity vector field of an instantaneous helical motion described by the pair $(\mathbf{c}_i, \bar{\mathbf{c}}_i)$, see Sect. 3.1. The velocity vectors are used for first order estimates of the new positions. Simultaneous variational subdivision (with uniform parametrization) on the K sequences of feature points amounts

parallel to A_i . For the computation of the Plücker coordinates $(\mathbf{a}_i, \bar{\mathbf{a}}_i)$ of the axis A_i and the pitch p_i from $(\mathbf{c}_i, \bar{\mathbf{c}}_i)$ see (10). This Euclidean transformation (25) brings the points $\mathbf{q}_i^1, \dots, \mathbf{q}_i^K$ to positions $\mathbf{q}_i^{1'}, \dots, \mathbf{q}_i^{K'}$ that are close to the tips $\mathbf{q}_i^1 + \mathbf{v}_i^1, \dots, \mathbf{q}_i^K + \mathbf{v}_i^K$ of the velocity vectors used in (18), see Fig. 5. We compute the new positions $\Phi'_{i|i+1}$ by

$$\Phi'_{i|i+1} := \mathbf{R}_i(\Phi_{i|i+1}) + \mathbf{t}_i. \quad (27)$$

Note that $\alpha_i = 0$, i.e., $\mathbf{c}_i = \mathbf{0}$, corresponds to a pure translation. Then $\mathbf{q}_i^{k'} = \mathbf{q}_i^k + \bar{\mathbf{c}}_i$ and thus $e_i^k := \|\mathbf{q}_i^{k'} - \mathbf{q}_i^k - \mathbf{v}_i^k\| = 0$. This expresses the fact that the velocity vector field of a pure translation coincides with the translation itself. If the pitch $p_i = 0$, we just apply a rotation through the angle α_i around the axis A_i .

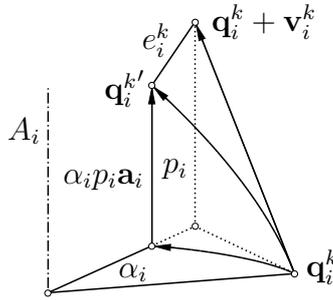


Fig. 5. New position $\mathbf{q}_i^{k'}$ of a point \mathbf{q}_i^k by applying the helical motion with the velocity vector field described by $(\mathbf{c}_i, \bar{\mathbf{c}}_i)$.

Remark 4 *The presented estimation of an instantaneous motion has also been used by Bourdet and Clément (1976, 1988) for the solution of a registration problem in Computer Vision. However, they directly transform the body (in an affine way) with the velocity vectors, and do not use the suggested correction via an appropriate helical motion.*

Then we use the new positions $\mathbf{q}_i^{k'}$ of the feature points as the starting positions for the next iteration step in the iterative optimization procedure. As a termination criterion for the iteration use the sum of squared distances between the positions of the feature points before and after the repositioning,

$$val := \sum_{i,k} \|\mathbf{q}_i^{k'} - \mathbf{q}_i^k\|^2. \quad (28)$$

If val falls below a certain threshold T , we terminate the optimization and insert the last obtained positions into the sequence of the fixed positions of the moving body. This new sequence of fixed positions is the input for the next subdivision step in our motion design algorithm.

Remark 5 *A summary of the unconstrained motion design algorithm is given in Alg. 2. Similarly to minimizing second forward differences in the motion*

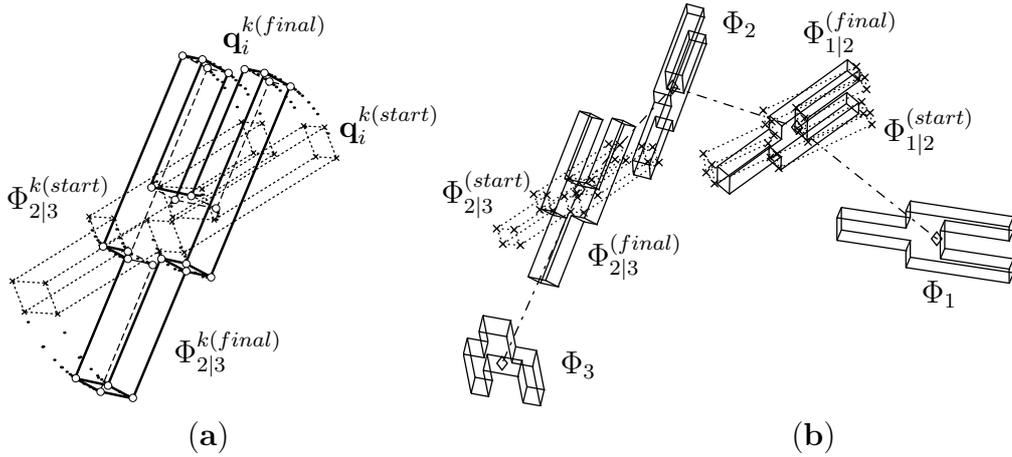


Fig. 6. One subdivision step in variational motion design: (a) Position insertion (\times), iterative position optimization (\cdot) and final position (\circ) of the feature points. (b) The dashed line connects the barycenters (\diamond) in their final positions.

design algorithm one generalizes non-uniform and approximating variational subdivision for motion design purposes. It is also possible to insert in each iteration step more than one intermediate position between known positions.

Fig. 7 shows an open and closed motion of a robot gripper arm interpolating 5 given positions.

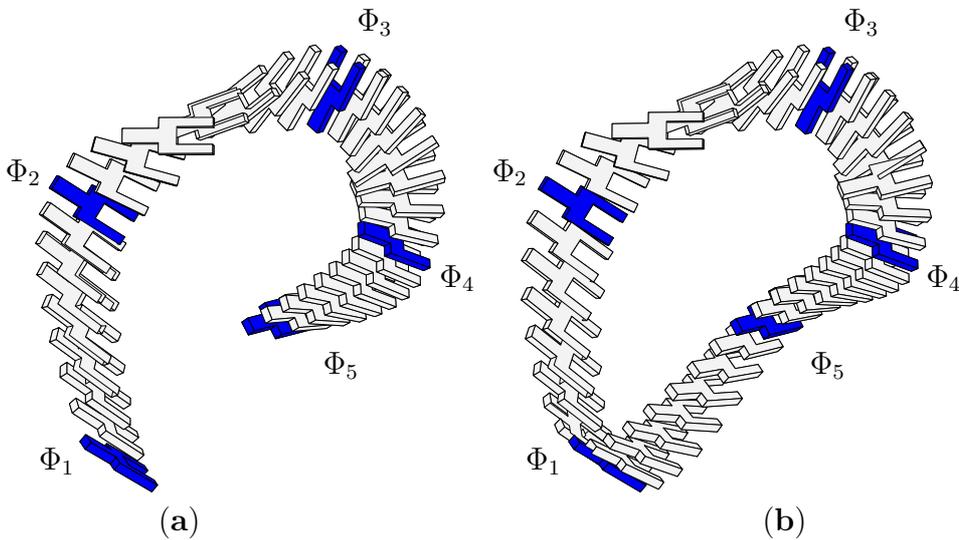


Fig. 7. Three variational subdivision steps for (a) open and (b) closed motion design to five input positions Φ_i .

Input: N_0 positions $\Phi_1^{(0)}, \dots, \Phi_{N_0}^{(0)}$ of a surface Φ , number of subdivision steps S , maximum number of optimization steps M , threshold T for optimization termination

for $l = 1$ **to** S

- (1) insert initial positions $\Phi_{i|i+1}^{(l-1)}$ between given positions $\Phi_i^{(l-1)}$ and $\Phi_{i+1}^{(l-1)}$, for $i = 1, \dots, N_{l-1} - 1$
 - compute translational parts $\mathbf{t}_i^{(l-1)}$ by (14)
 - compute rotational parts $\mathbf{R}_i^{(l-1)}$ by (15)
 - apply transformation (13)
 - (2) **while** $j < M$ **and** $val > T$ **do**
 - compute $(\mathbf{c}_i, \bar{\mathbf{c}}_i)$ by minimizing (18)
 - transform $\Phi_{i|i+1}^{(l-1)}$ via (27) to new position $\Phi'_{i|i+1}{}^{(l-1)}$
 - compute val via (28)
 - $\Phi_{i|i+1}^{(l-1)} = \Phi'_{i|i+1}{}^{(l-1)}$
 - $j = j + 1$

end
 - (3) $\Phi_{2i-1}^{(l)} := \Phi_i^{(l-1)}, i = 1, \dots, N_{l-1}$
 $\Phi_{2i}^{(l)} := \Phi_{i|i+1}^{(l-1)}, i = 1, \dots, N_{l-1} - 1$
- end**

Output: $N_S := (N_0 - 1)2^S + 1$ positions $\Phi_1^{(S)}, \dots, \Phi_{N_S}^{(S)}$ of the surface Φ

Alg. 2: Variational subdivision algorithm for unconstraint motion design.

4.4 Higher-dimensional viewpoint and dependency on feature points

In this section we give a higher-dimensional viewpoint of the presented unconstraint motion design algorithm. Furthermore we show that our algorithm gives the same result for every choice of feature points with the same barycenter and the same inertia tensor.

An affine transformation α in Euclidean three-space E^3 is given by

$$\mathbf{x} \mapsto \mathbf{a} + \mathbf{A}\mathbf{x}, \quad \mathbf{a} \in \mathbb{R}^3, \quad \mathbf{A} \in \mathbb{R}^{3 \times 3}, \quad \det \mathbf{A} \neq 0. \quad (29)$$

We denote the 12-dimensional space of affine transformations by A^{12} , and the 6-dimensional manifold of Euclidean transformations (i.e., for the matrix \mathbf{A} in (29) we have $\mathbf{A}^T \mathbf{A} = \mathbf{A} \mathbf{A}^T = \mathbf{I}$) by M^6 . We define a Euclidean metric in A^{12} via the sum of squared distances between feature points of the moving body in homologous positions in E^3 .

Definition 6 *Let α, β be two affine maps that are applied to a Euclidean body*

Φ , which is represented by feature points $\mathbf{f}^1, \dots, \mathbf{f}^K$. We define the distance $d(\alpha(\Phi), \beta(\Phi))$ between $\alpha(\Phi)$ and $\beta(\Phi)$ by

$$d^2(\alpha(\Phi), \beta(\Phi)) := \sum_{i=1}^K \|\alpha(\mathbf{f}^i) - \beta(\mathbf{f}^i)\|^2. \quad (30)$$

Remark 7 If a moving body is represented by a finite number of feature points $\mathbf{f}^1, \dots, \mathbf{f}^K$ of unit point masses, then the coordinate matrix \mathbf{J} of the inertia tensor is given by

$$\mathbf{J} = \sum_{i=1}^K \mathbf{f}^i (\mathbf{f}^i)^T. \quad (31)$$

Proposition 8 The Euclidean metric in A^{12} , defined via the squared distance $d^2(\alpha(\Phi), \beta(\Phi))$, only depends on the barycenter and on the inertia tensor of the rigid body Φ .

PROOF. Let $\mathbf{f}^1, \dots, \mathbf{f}^K$ be a feature point representation of a Euclidean body Φ , and let α, β be two affine maps given by (29). We rewrite $d^2(\alpha(\Phi), \beta(\Phi))$ from (30) using (29) as follows,

$$\begin{aligned} d^2(\alpha(\Phi), \beta(\Phi)) &= \sum_{i=1}^K \|\alpha(\mathbf{f}^i) - \beta(\mathbf{f}^i)\|^2 \\ &= \sum_{i=1}^K (\mathbf{a} + \mathbf{A}\mathbf{f}^i - \mathbf{b} - \mathbf{B}\mathbf{f}^i)^T (\mathbf{a} + \mathbf{A}\mathbf{f}^i - \mathbf{b} - \mathbf{B}\mathbf{f}^i) \\ &:= \sum_{i=1}^K (\mathbf{d} + \mathbf{D}\mathbf{f}^i)^T (\mathbf{d} + \mathbf{D}\mathbf{f}^i) \\ &= \sum_{i=1}^K (\mathbf{d}^T \mathbf{d} + 2\mathbf{d}^T \mathbf{D}\mathbf{f}^i + \mathbf{f}^{iT} \mathbf{D}^T \mathbf{D}\mathbf{f}^i) \\ &= K\mathbf{d}^T \mathbf{d} + 2\mathbf{d}^T \mathbf{D} \sum_{i=1}^K \mathbf{f}^i + \sum_{i=1}^K \mathbf{f}^{iT} \mathbf{Q}\mathbf{f}^i \end{aligned} \quad (32)$$

where $\mathbf{d} := \mathbf{a} - \mathbf{b}$, $\mathbf{D} := \mathbf{A} - \mathbf{B}$ and $\mathbf{Q} := \mathbf{D}^T \mathbf{D}$. The second term in (32) contains the barycenter of the point cloud $\mathbf{f}^1, \dots, \mathbf{f}^K$ and the last term contains the components $\sum_i f_j^i f_k^i$ of the inertia tensor described by the matrix \mathbf{J} . Thus for fixed α and β , $d^2(\alpha(\Phi), \beta(\Phi))$ only depends on the barycenter and on the inertia tensor of Φ . Note that (32) can be rewritten as

$$d^2(\alpha, \beta) = (\hat{\mathbf{a}} - \hat{\mathbf{b}})^T \mathbf{M} (\hat{\mathbf{a}} - \hat{\mathbf{b}}) \quad (33)$$

with a positive definite 12×12 matrix \mathbf{M} . Then (33) is the familiar form of

a Euclidean metric. The vectors $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ collect the twelve parameters (\mathbf{a}, \mathbf{A}) and (\mathbf{b}, \mathbf{B}) of the affine transformations α and β . \square

Thus, the Euclidean metric we have defined is the same for all choices of feature points on Φ with the same barycenter $\mathbf{s} := 1/N \sum_{i=1}^K \mathbf{f}^i$ and the same inertia tensor. By a well-known result from mechanics we can replace the feature points $\mathbf{f}^1, \dots, \mathbf{f}^K$ by the six special points

$$\pm \mathbf{f}^i := \mathbf{s} \pm \sqrt{\frac{\lambda_i}{2}} \mathbf{e}_i, \quad i = 1, 2, 3, \quad (34)$$

where $\lambda_1, \lambda_2, \lambda_3$ and $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ are the eigenvalues and eigenvectors of \mathbf{J} .

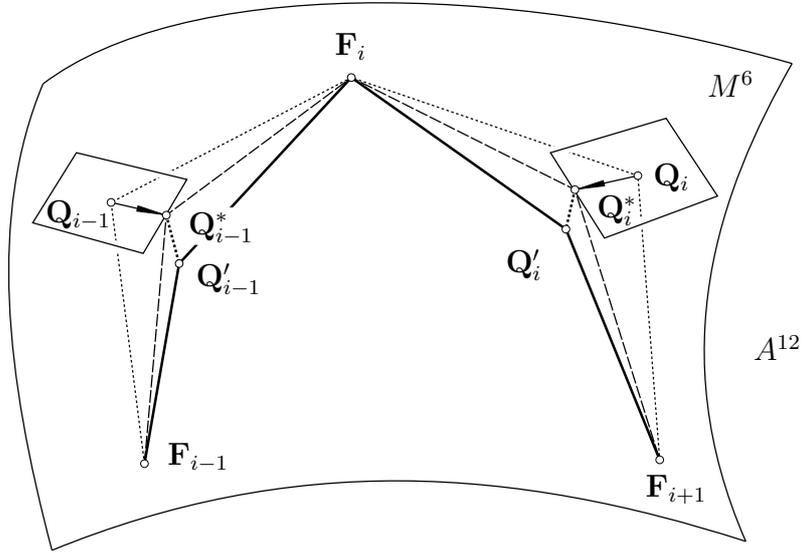


Fig. 8. The motion design algorithm sketched in A^{12} .

We compare the motion design algorithm presented in this paper to a previously discussed algorithm: The algorithm for the design of a rigid body motion presented by Hofer et al. (2003) first computes an affine motion $\alpha(t)$ (i.e., via a variational approach a curve in A^{12}) that interpolates M^6 in given points \mathbf{F}_i . Then $\alpha(t)$ is projected orthogonally onto M^6 . This orthogonality is understood in the Euclidean metric defined in (33).

The approach we present in this paper designs via variational subdivision a curve on M^6 : Between given points $\mathbf{F}_i \in M^6$ we insert initial points $\mathbf{Q}_i \in M^6$ to which we attach unknown vectors \mathbf{V}_i in the 6-dimensional tangent space of M^6 at \mathbf{Q}_i (parameterized with help of $(\mathbf{c}_i, \bar{\mathbf{c}}_i)$). Then we minimize a quadratic energy functional for the points $\dots, \mathbf{F}_i, \mathbf{Q}_i^* := \mathbf{Q}_i + \mathbf{V}_i, \mathbf{F}_{i+1}, \dots$. Since the such computed new positions \mathbf{Q}_i^* are in A^{12} but not on M^6 we replace them

by positions \mathbf{Q}'_i on M^6 close to them. In the algorithm this is done by using the helical motion with the velocity vector field $(\mathbf{c}_i, \bar{\mathbf{c}}_i)$. From a higher-dimensional viewpoint this corresponds to an orthogonal projection of $\mathbf{Q}_i^* \in A^{12}$ to $\mathbf{Q}'_i \in M^6$ via the metric (33). Now Prop. 8 says that this projection is the same for every choice of feature points with the same barycenter and same inertia tensor. Now this two step procedure (insertion of \mathbf{Q}_i^* and orthogonal projection onto \mathbf{Q}'_i) is repeated until the quadratic energy functional is minimized. Then we continue with the next subdivision step.

Remark 9 *Note that our algorithm for motion design is a special case of variational curve design (via subdivision) on arbitrary k -dimensional surfaces in \mathbb{R}^d . One example has been given in Sect. 2.1 for a 2-dimensional surface in \mathbb{R}^3 , another example follows in Sect. 5, where a 5-dimensional surface contained in M^6 arises from a contact condition on the designed motion.*

4.5 Conjecture concerning smoothness of the motion

The variational interpolatory subdivision scheme we use in our approach produces at least C^2 -curves, as proved by Kobbelt (1996). However, we have slightly modified this scheme and apply it to several input polygons simultaneously. Furthermore, we do not reposition the feature points by the velocity vectors used in the computation – which would lead to affinely distorted copies of our moving body – but with the underlying helical motion. Therefore it is more difficult to theoretically analyze the smoothness of the resulting motion. However, numerical tests (the second numerical derivatives of the paths of the feature points are ‘continuous’) suggest that the computed motions are C^2 .

5 Geometric design of motions constrained by a contacting surface pair

In this section we present an algorithm that solves the following problem: Given are N positions Φ_i of a moving surface Φ in contact with a fixed surface Ψ , see Fig. 9. Contact means that the surface normals of both surfaces coincide in the contact points. The objective is to compute a motion such that Φ *glides* on Ψ and interpolates (or approximates) the given positions Φ_i .

The gliding motion of a surface Φ on a surface Ψ is a five-parameter motion. At every instance, the gliding motion of two surfaces has an instantaneous path normal complex which contains the common normal of the gliding surface pair. Up to first order this is the only constraint on a gliding surface pair, see e.g. Pottmann and Ravani (2000). Thus we extend the subdivision algorithm for

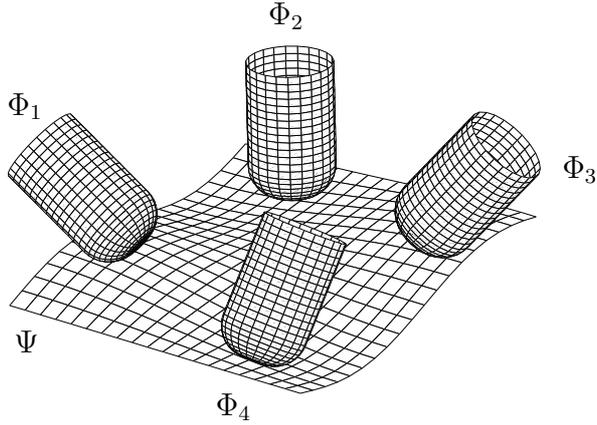


Fig. 9. The fixed surface Ψ and the input positions Φ_i of the moving surface Φ in contact with Ψ .

motion design presented in Sect. 4 to the design of gliding motions by adding this gliding constraint.

The instantaneous motions used in the optimization procedure (18) are described by pairs $(\mathbf{c}_i, \bar{\mathbf{c}}_i)$. The common normal n_i of the two surfaces Ψ and Φ_i in the instantaneous contact point has normalized Plücker coordinates $(\mathbf{n}_i, \bar{\mathbf{n}}_i)$. The constraint, that n_i is contained in the instantaneous path normal complex of the motion, follows from (12) and thus reads as

$$\mathbf{c}_i \cdot \bar{\mathbf{n}}_i + \bar{\mathbf{c}}_i \cdot \mathbf{n}_i = 0, \quad i = 1, \dots, N - 1. \quad (35)$$

These linear constraints are added to the quadratic objective function (18) and the such obtained energy functional is minimized. The optimization is again based on a feature point representation of the moving surface Φ , see Sect. 4.4.

5.1 Outline of the algorithm

Before we discuss the individual steps of the algorithm in more detail we give an outline of the subdivision algorithm for gliding motion design:

- (1) **Position insertion:** Insert initial intermediate positions of the moving surface in contact with the fixed surface between each two given contact positions.
 - (a) Insert with the algorithm for variational design of a surface curve (Sect. 2.1) new intermediate contact points on both, the fixed and the moving surface. Compute the surface normals in these new contact points. Let the new points and normals with the same index

- correspond to each other.
- (b) Insert Euclidean copies $\Phi_{i|i+1}$ such that the surface normals of fixed and moving surface coincide in the new contact points.
 - (c) Apply an iterative position optimization algorithm to all $\Phi_{i|i+1}$ simultaneously by using velocity vector fields of pure rotations around the common surface normals in the new contact points. This gives good starting positions for the following optimization with gliding constraints.
- (2) **Position optimization with gliding constraint:**
- (a) Minimize the quadratic energy functional (18) with the linear constraint (35) and apply the resulting helical motions to the inserted copies $\Phi_{i|i+1}$ to get $\Phi'_{i|i+1}$.
 - (b) Note that after step (a) the moving surface might not be in contact with the fixed surface. Therefore we compute the common normals of Ψ and $\Phi'_{i|i+1}$ and translate each $\Phi'_{i|i+1}$ such that it is again in contact with Ψ .
 - (c) Repeat steps (a) and (b) until the constraint objective function is minimized. Then insert the final positions into the sequence of fixed positions of Φ and use these new $2N - 1$ copies as input to the next subdivision step.

Remark 10 *Note that the optimization is computed using a feature point representation of the moving surface. But once we have found the transformations that perform the relocations of the feature points, we can also apply these transformations to the moving surface itself (and vice versa).*

5.2 Position insertion

The input positions Φ_i ($i = 1, \dots, N$) of the moving surface Φ are in contact with the fixed surface Ψ , i.e., in the contact point of Ψ and Φ_i the surface normals coincide. Between each two adjacent input positions Φ_i and Φ_{i+1} we insert an initial intermediate position $\Phi_{i|i+1}$ of the moving surface Φ in such a way that $\Phi_{i|i+1}$ is in contact with Ψ and both surfaces have the same surface normal in the contact point. The $\Phi_{i|i+1}$ are then the start positions for an optimization with gliding constraints.

For the insertion of new contact points we apply one subdivision step of the algorithm for variational surface curve design (presented in Sect. 2.1) to the given contact points on both surfaces, the fixed and the moving one, separately. In these new corresponding contact points we compute the surface normal for each surface. Then we insert copies $\Phi_{i|i+1}$ of the moving surface such that the inserted contact points of the moving surface are mapped to the inserted contact points of the fixed surface and that the surface normals of Ψ and Φ

in those points coincide.

The such obtained positions $\Phi_{i|i+1}$ are not unique. For every position there is still one degree of freedom, the rotation around the common surface normal. We use this degree of freedom for every $\Phi_{i|i+1}$ for a simultaneous optimization with a variant of the variational subdivision algorithm for motion design presented in Sect. 4. Each velocity vector field is restricted to coincide up to first order with a *pure rotation around the common normal* in the contact point of fixed and moving surface. This reduces the six free parameters $(\mathbf{c}_i, \bar{\mathbf{c}}_i)$ for every moving position in (18) to only one, the rotation angle $\gamma_i := \|\mathbf{c}_i\|$. Since for a pure rotation the pitch p is zero, we conclude from (10) that $(\mathbf{c}_i, \bar{\mathbf{c}}_i)$ are of the form

$$(\mathbf{c}_i, \bar{\mathbf{c}}_i) = \|\mathbf{c}_i\|(\mathbf{n}_i, \bar{\mathbf{n}}_i), \quad i = 1, \dots, N - 1. \quad (36)$$

The objective function (18) with the $6(N - 1)$ unknowns $\mathbf{c}_1, \bar{\mathbf{c}}_1, \dots, \mathbf{c}_{N-1}, \bar{\mathbf{c}}_{N-1}$ now reduces to the following objective function, which is quadratic in the $N - 1$ unknowns $\gamma_1, \dots, \gamma_{N-1}$,

$$\begin{aligned} F_2(\gamma_1, \dots, \gamma_{N-1}) &= \sum_{k=1}^K \sum_{i=1}^{N-1} \|\mathbf{f}_i^k - 2(\mathbf{q}_i^k + \mathbf{v}_i^k) + \mathbf{f}_{i+1}^k\|^2 \\ &+ \sum_{i=1}^{N-2} \|(\mathbf{q}_i^k + \mathbf{v}_i^k) - 2\mathbf{f}_{i+1}^k + (\mathbf{q}_{i+1}^k + \mathbf{v}_{i+1}^k)\|^2, \end{aligned} \quad (37)$$

where

$$\mathbf{v}_i^k := \mathbf{v}(\mathbf{q}_i^k) = \gamma_i(\bar{\mathbf{n}}_i + \mathbf{n}_i \times \mathbf{q}_i^k). \quad (38)$$

The minimization of (37) is found by solving the linear system of equations

$$\mathbf{A}^r \mathbf{\Gamma} = \mathbf{B}^r \quad (39)$$

We collect the unknowns in $\mathbf{\Gamma} = (\gamma_1, \dots, \gamma_{N-1})$ and derive for $i = 1, \dots, N - 1$ the i -th entry of the vector \mathbf{B}^r as

$$\mathbf{B}^r(i) = \mathbf{B}(6i - 5, \dots, 6i - 3) \cdot \mathbf{n}_i + \mathbf{B}(6i - 2, \dots, 6i) \cdot \bar{\mathbf{n}}_i, \quad (40)$$

where \mathbf{B} is given by (20). The entries on the main diagonal of the tridiagonal $(N - 1) \times (N - 1)$ *symmetric* matrix \mathbf{A}^r are given by

$$a_{jj} = 5K\bar{\mathbf{n}}_j^2 + 10\left(\sum_k \mathbf{q}_j^k\right) \cdot (\bar{\mathbf{n}}_j \times \mathbf{n}_j) + 5\left(\sum_k \mathbf{q}_{jj}^{k\circ}\right) \cdot \mathbf{n}_{jj}^* - 10\left(\sum_k \mathbf{q}_{jj}^{k*}\right) \cdot \mathbf{n}_{jj}^*,$$

$$a_{ii} = 6K\bar{\mathbf{n}}_i^2 + 12\left(\sum_k \mathbf{q}_i^k\right) \cdot (\bar{\mathbf{n}}_i \times \mathbf{n}_i) + 6\left(\sum_k \mathbf{q}_{ii}^{k\diamond}\right) \cdot \mathbf{n}_{ii}^* - 12\left(\sum_k \mathbf{q}_{ii}^{k\star}\right) \cdot \mathbf{n}_{ii}^*,$$

for $j = 1, N-1$ and $i = 2, \dots, N-2$. The elements on the upper diagonal are the real numbers

$$\begin{aligned} a_{i,i+1} = & K(\bar{\mathbf{n}}_i \cdot \bar{\mathbf{n}}_{i+1}) + \left(\sum_k \mathbf{q}_{i+1}^k\right) \cdot (\bar{\mathbf{n}}_i \times \mathbf{n}_{i+1}) + \left(\sum_k \mathbf{q}_i^k\right) \cdot (\bar{\mathbf{n}}_{i+1} \times \mathbf{n}_i) \\ & + \left(\sum_k \mathbf{q}_{i,i+1}^{k\star}\right) \cdot \mathbf{n}_{i,i+1}^\diamond - \left(\sum_k \mathbf{q}_{i,i+1}^{k\star}\right) \cdot \mathbf{n}_{i+1,i}^* - \left(\sum_k \mathbf{q}_{i+1,i}^{k\star}\right) \cdot \mathbf{n}_{i,i+1}^*, \end{aligned} \quad (41)$$

for $i = 1, \dots, N-2$. The superscripts $*$, \star , \diamond denote special vectors, here defined in terms of $\mathbf{n}_i = (n_{i1}, n_{i2}, n_{i3})^T$ and $\mathbf{n}_j = (n_{j1}, n_{j2}, n_{j3})^T$,

$$\mathbf{n}_{ij}^* := \begin{pmatrix} n_{i1}n_{j1} \\ n_{i2}n_{j2} \\ n_{i3}n_{j3} \end{pmatrix}, \quad \mathbf{n}_{ij}^\star := \begin{pmatrix} n_{i1}n_{j3} \\ n_{i2}n_{j1} \\ n_{i3}n_{j2} \end{pmatrix}, \quad \mathbf{n}_{ij}^\diamond := \begin{pmatrix} n_{i2}n_{j2} + n_{i3}n_{j3} \\ n_{i1}n_{j1} + n_{i3}n_{j3} \\ n_{i1}n_{j1} + n_{i2}n_{j2} \end{pmatrix}. \quad (42)$$

The solution of (39) gives us $\gamma_1, \dots, \gamma_{N-1}$ and to each position $\Phi_{i|i+1}$ we apply the rotations through angles γ_i around the common surface normals of $\Phi_{i|i+1}$ and Ψ . Then we iterate the whole procedure until the minimum of the objective function (37) is reached. This gives the initial intermediate positions $\Phi_{i|i+1}$ ($i = 1, \dots, N-1$) that are used as starting values for the constraint optimization described in Sect. 5.3.

5.3 Position optimization with gliding constraint

In this step of the algorithm we use the above computed positions of the moving surface as starting values for an iterative optimization algorithm. As in Sect. 4 we denote the feature points in the position $\Phi_{i|i+1}$ by \mathbf{q}_i^k and attach vectors

$$\mathbf{v}_i^k := \mathbf{v}(\mathbf{q}_i^k) = \bar{\mathbf{c}}_i + \mathbf{c}_i \times \mathbf{q}_i^k \quad (43)$$

to them. The \mathbf{v}_i^k again belong to the linear velocity vector field of an instantaneous helical motion described by the pair $(\mathbf{c}_i, \bar{\mathbf{c}}_i)$, see Sect. 3.1. The velocity vectors are used for first order estimates of the new positions. The gliding motion of two surfaces has an instantaneous *path normal complex* which contains the common normal of the gliding surface pairs Ψ and $\Phi_{i|i+1}$,

$$\mathbf{c}_i \cdot \bar{\mathbf{n}}_i + \bar{\mathbf{c}}_i \cdot \mathbf{n}_i = 0, \quad i = 1, \dots, N-1. \quad (44)$$

Up to first order this is the only gliding constraint. Thereby, $(\mathbf{c}_i, \bar{\mathbf{c}}_i)$ are the yet unknown representations of the velocity vector fields and $(\mathbf{n}_i, \bar{\mathbf{n}}_i)$ are the Plücker coordinates of the contact normals in the initial intermediate positions. This gives $N - 1$ *linear* constraints, one for each inserted position of the moving body. Using (44) and $\|\mathbf{n}\| = 1$ one can eliminate $3(N - 1)$ unknowns, since

$$\bar{\mathbf{c}}_i = -(\mathbf{c}_i \cdot \bar{\mathbf{n}}_i)\mathbf{n}_i, \quad i = 1, \dots, N - 1. \quad (45)$$

Substitution of (45) in the objective function (18) results in solving a quadratic objective function for the remaining $3(N - 1)$ unknowns \mathbf{c}_i .

However, we introduce Lagrange multipliers $\lambda_1, \dots, \lambda_{N-1}$, since this allows us to describe the linear system of equations, that has to be solved, in terms of that of Sect. 4.3. The Lagrange function L consists of the quadratic objective function F_2 from (18) and λ_i times the linear constraints (44),

$$L(\mathbf{c}_1, \bar{\mathbf{c}}_1, \dots, \mathbf{c}_{N-1}, \bar{\mathbf{c}}_{N-1}, \lambda_1, \dots, \lambda_{N-1}) = F_2 + \sum_{i=1}^{N-1} \lambda_i (\mathbf{c}_i \cdot \bar{\mathbf{n}}_i + \bar{\mathbf{c}}_i \cdot \mathbf{n}_i). \quad (46)$$

Computing the partial derivatives of L with respect to $\mathbf{C}_i := (\mathbf{c}_i, \bar{\mathbf{c}}_i)$ and λ_i ,

$$\frac{\partial L}{\partial \mathbf{C}_i} = \mathbf{o}, \quad \frac{\partial L}{\partial \lambda_i} = 0, \quad i = 1, \dots, N - 1, \quad (47)$$

leads to a system of $7(N - 1)$ linear equations

$$\mathbf{A}^c \cdot \mathbf{C}^c = \mathbf{B}^c. \quad (48)$$

We find $\mathbf{B}^c, \mathbf{C}^c, \mathbf{A}^c$ by extending $\mathbf{B}, \mathbf{C}, \mathbf{A}$ from (20), (21), (22) respectively in the following way. The vector \mathbf{B}^c is equal to the vector \mathbf{B} extended by $(N - 1)$ zeros, and the vector of unknowns \mathbf{C}^c is derived from \mathbf{C} by adding the $N - 1$ Lagrange multipliers λ_i at the end,

$$\mathbf{B}^c = (\mathbf{B}^T, \underbrace{0, \dots, 0}_{N-1})^T, \quad \mathbf{C}^c = (\mathbf{C}^T, \lambda_1, \dots, \lambda_{N-1})^T. \quad (49)$$

The symmetric matrix \mathbf{A}^c is a block matrix with \mathbf{A} from (22) as an $6(N - 1) \times 6(N - 1)$ block in the upper left corner, the $6(N - 1) \times (N - 1)$ block matrix \mathbf{N} in the upper right corner, its transpose \mathbf{N}^T and the $(N - 1) \times (N - 1)$ zero

matrix $\mathbf{0}$ in the lower two blocks,

$$\mathbf{A}^c = \left(\begin{array}{c|c} \mathbf{A} & \mathbf{N} \\ \hline \mathbf{N}^T & \mathbf{0} \end{array} \right) \quad \text{with} \quad \mathbf{N}^T = \begin{pmatrix} \bar{\mathbf{n}}_1^T & \mathbf{n}_1^T & & \\ & & \ddots & \ddots \\ & & & \bar{\mathbf{n}}_{N-1}^T & \mathbf{n}_{N-1}^T \end{pmatrix}. \quad (50)$$

Once we have solved (48) we use the helical motions corresponding to $(\mathbf{c}_i, \bar{\mathbf{c}}_i)$ to separately reposition each intermediate position $\Phi_{i|i+1}$ of the moving surface and of the feature points \mathbf{q}_i^k . Note that after we have applied the helical motions, the resulting positions $\Phi'_{i|i+1}$ might not be in contact with Ψ anymore. Therefore we have to perform the following position correction, which we just briefly outline: Compute the common normal (cf. e.g. Johnson and Cohen (1998) and Sohn et al. (2002) and the references therein) of Ψ and $\Phi_{i|i+1}$ which intersects the two surfaces in points $\mathbf{p}_{i|i+1,\Psi}$ and $\mathbf{p}_{i|i+1,\Phi}$. Then translate $\Phi_{i|i+1}$ by the vector

$$\mathbf{t}_{i|i+1} := \mathbf{p}_{i|i+1,\Psi} - \mathbf{p}_{i|i+1,\Phi}. \quad (51)$$

Now the fixed and the copies of the moving surface are again in contact and we continue with the next subdivision step of the gliding motion design algorithm.

Remark 11 *We have presented the subdivision algorithms for smooth and gliding motion design such that the computed motion interpolates the given positions. This algorithm is summarized in Alg. 3. With the same approach, using a generalization of approximating variational subdivision we can design smooth and gliding motions that approximate given input positions.*

5.4 Examples

The numerical experiments have been performed with a Matlab implementation on a 1.8 GHZ Pentium 4 with 1GB RAM. Our first example simulates a toroidal cutter (for a definition see Hoschek and Lasser (1993)) Φ gliding on a wave surface Ψ . Fig. 10 shows the inserted copies of the moving surface Φ after the first to fourth subdivision step. Fig. 11 shows the contact curves on Ψ and Φ respectively. The second example is a toroidal cutter Φ gliding on a ruled surface Ψ . Fig. 13 shows the computed positions after the third subdivision step. The third example is a toroidal cutter Φ gliding on an ellipsoid Ψ . Fig. 12 shows the computed positions after the fourth subdivision step. Table 2 shows some numerical results for the computed gliding motions: For each of the three examples we have computed four subdivision steps. We list the number of calls of (37) and (46) in every subdivision step and the total

Input: N_0 positions $\Phi_1^{(0)}, \dots, \Phi_{N_0}^{(0)}$ of a surface Φ in point contact with a surface Ψ , number of subdivision steps S , maximum number of optimization steps M , threshold T for optimization termination

for $l = 1$ **to** S

- (1) insert initial positions $\Phi_{|i+1}^{(l-1)}$ between given positions $\Phi_i^{(l-1)}$ and $\Phi_{i+1}^{(l-1)}$ in point contact with Ψ for $i = 1, \dots, N_{l-1} - 1$
 - apply one subdivision step of the variational surface curve algorithm described in Sect. 2.1 to the contact points on Φ and Ψ
 - compute the surface normals of Φ and Ψ in the new contact points
 - insert $\Phi_{|i+1}^{(l-1)}$ such that $\Phi_{|i+1}^{(l-1)}$ and Ψ touch each other in the new contact point and that the surface normals coincide

while $j_1 < M$ **and** $val_1 > T$ **do**

 - compute γ_i by minimizing (37)
 - rotate $\Phi_{|i+1}^{(l-1)}$ around the common surface normal to new position $\Phi_{|i+1}'^{(l-1)}$
 - compute $val_1 = \sum_{i,k} \|\mathbf{q}_i^{k'} - \mathbf{q}_i^k\|^2$
 - $\Phi_{|i+1}^{(l-1)} = \Phi_{|i+1}'^{(l-1)}$
 - $j_1 = j_1 + 1$

end
 - (2) Optimize positions $\Phi_{|i+1}'^{(l-1)}$ with gliding constraint (35)

while $j_2 < M$ **and** $val_2 > T$ **do**

 - compute $(\mathbf{c}_i, \bar{\mathbf{c}}_i)$ by minimizing (46)
 - transform $\Phi_{|i+1}'^{(l-1)}$ via the helical motion corresponding to $(\mathbf{c}_i, \bar{\mathbf{c}}_i)$ to new position $\Phi_{|i+1}''^{(l-1)}$
 - compute $val_2 = \sum_{i,k} \|\mathbf{q}_i^{k'} - \mathbf{q}_i^k\|^2$
 - compute common normal of $\Phi_{|i+1}''^{(l-1)}$ and Ψ and translate $\Phi_{|i+1}''^{(l-1)}$ by (51)
 - $\Phi_{|i+1}^{(l-1)} = \Phi_{|i+1}''^{(l-1)}$
 - $j_2 = j_2 + 1$

end
 - (3) $\Phi_{2i-1}^{(l)} := \Phi_i^{(l-1)}, i = 1, \dots, N_{l-1}$
 $\Phi_{2i}^{(l)} := \Phi_{|i+1}^{(l-1)}, i = 1, \dots, N_{l-1} - 1$
- end**

Output: $N_S := (N_0 - 1)2^S + 1$ positions $\Phi_1^{(S)}, \dots, \Phi_{N_S}^{(S)}$ of the moving surface Φ in contact with the fixed surface Ψ

Alg. 3: Variational subdivision algorithm for design of a motion constraint by a contacting surface pair.

computation time for the respective operation (for the ‘wave’ example, (37) has been called 33, 10, 7, 7 times in the 1, 2, 3, 4 subdivision step and the total computation time for these 57 calls was 0.74s. If the input positions are far apart (as in the case of the wave and the ellipsoid example), then we insert – by the first two subdivision steps – three intermediate positions for which we only employ (37), and then from the third subdivision step onwards we also employ the constraint optimization (46).

| | Wave | Ruled surface | Ellipsoid |
|-----------------------|------------------|------------------|------------------|
| input positions | 4 | 5 | 4 |
| calls/total time (37) | 33,10,7,7/0.74s | 3,2,2,2/0.21s | 14,9,8,8/0.68s |
| calls/total time (46) | -, -, 2, 2/0.13s | 2, 2, 2, 2/0.36s | -, -, 4, 4/0.37s |

Table 2

Numerical results for the gliding motion shown in Fig. 10, 12 and 13.

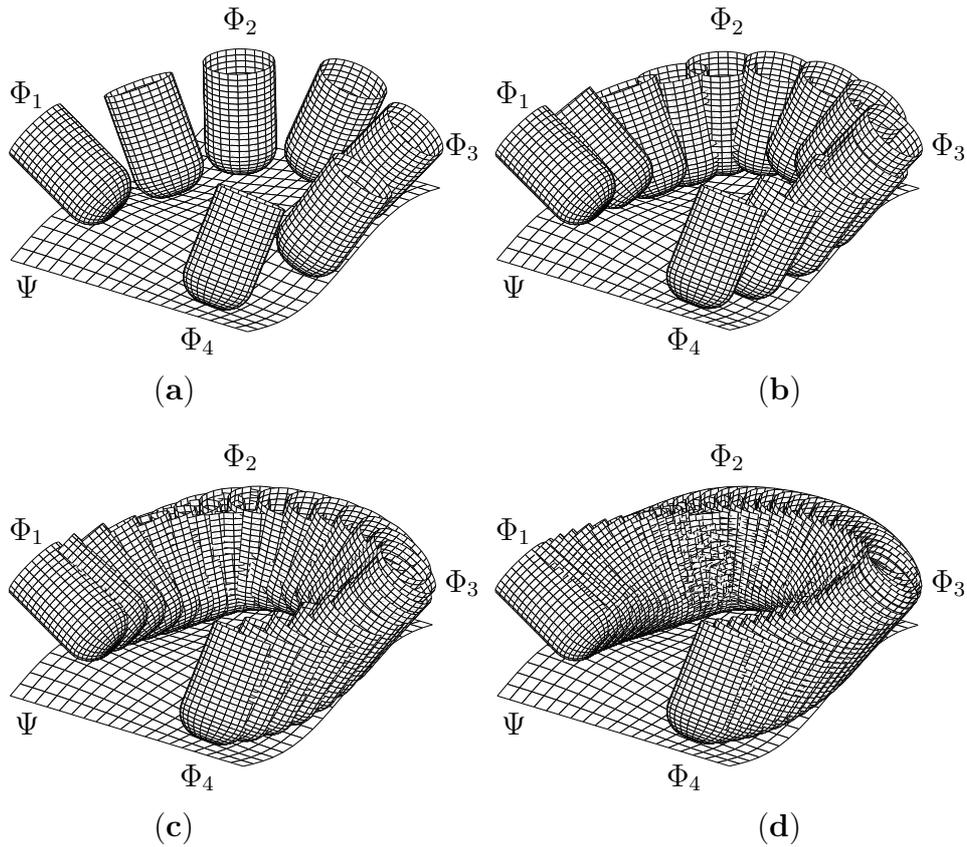


Fig. 10. Toroidal cutter Φ gliding on a wave surface Ψ : (a), ..., (d) show the positions of Φ after the first to fourth subdivision step of the gliding motion design algorithm.

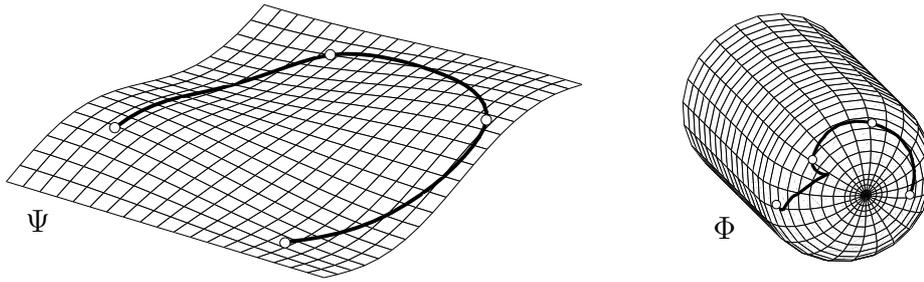


Fig. 11. The contact curve on fixed and moving surface.

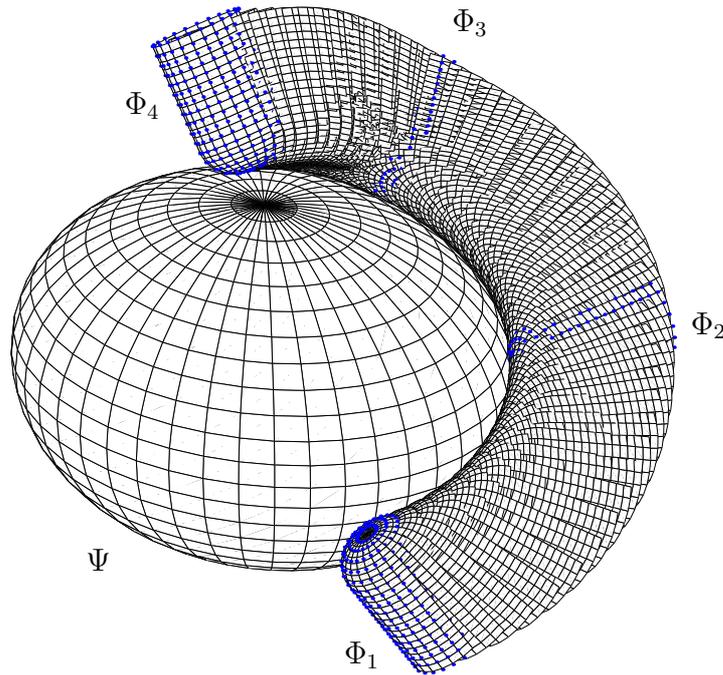


Fig. 12. Toroidal cutter Φ gliding on an ellipsoid Ψ .

6 Conclusions and future work

The main contribution of this paper is a subdivision algorithm for geometric design of motions constraint by a contacting surface pair. The presented algorithm uses instantaneous kinematics, line geometry, and variational subdivision algorithms. We have first presented a motion design algorithm for unconstrained motion design and then we included the constraint of a contacting surface pair to get the main algorithm. In this paper we have also extended known interpolatory variational subdivision algorithms for curve design to special cases of variational curve design on k -dimensional surfaces in \mathbb{R}^d .

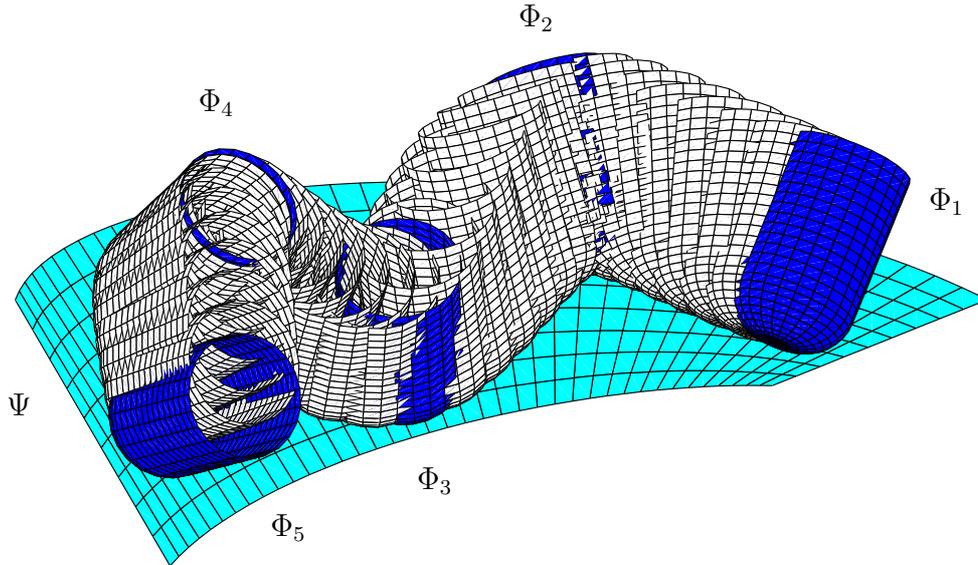


Fig. 13. Toroidal cutter Φ gliding on a ruled surface Ψ .

The results can be extended to the *design of smooth motions in the presence of obstacles*. There are many algorithms to design smooth motions based on techniques from Computer Aided Geometric Design, and a variety of solutions for the avoidance of obstacles based on Computational Geometry. Whereas the latter approaches do not deal with spline-type smooth motions, the former are lacking an efficient treatment of obstacles. Using the presented subdivision methods for (gliding) motion design, the problem of obstacle avoidance can be investigated. This might be the subject of future research.

7 Acknowledgements

The authors would like to thank the anonymous reviewers for their useful comments. This work was supported, in part, by the California Department of Transportation through the basic research component of the AHMCT research center at UC Davis. This research has also been carried out, in part, in connection with projects P13938MAT and P16002-N05, supported by the Austrian Science Fund (FWF).

References

- Belta, C., Kumar, V. (2002) An SVD-Projection Method for Interpolation on $SE(3)$. *IEEE Trans. on Robotics and Automation* **18**, 334–345.
- Bourdet, P., Clément, A. (1976) Controlling a complex surface with a 3 axis measuring machine. *Annals of the CIRP* **25**, 359–361.

- Bourdet, P., Clément, A. (1988) A study of optimal-criteria identification based on the small-displacement screw model. *Annals of the CIRP* **37**, 503–506.
- Fang, Y.C., Hsieh, C.C., Kim, M.J., Chang, J.J., Woo, T.C. (1998), Real time motion fairing with unit quaternions. *Computer-Aided Design*, **30**, 191–198.
- Fletcher, R. (2000), *Practical Methods of Optimization*, 2nd edition, Wiley.
- Halperin, D., Kavraki, L., Latombe, J.C. (1997) Robotics. In: Handbook of Discrete and Computational Geometry. J. E. Goodman and J. O'Rourke, eds., CRC Press, New York, 775–778.
- Hofer, M., Pottmann, H., Ravani, B. (2002) Subdivision algorithms for motion design based on homologous points. In: Lenarcic, J., Thomas, F., eds., Advances in Robot Kinematics – Theory and Applications, Kluwer Academic Publishers, 235-244.
- Hofer, M., Pottmann, H., Ravani, B. (2003) From curve design algorithms to the design of rigid body motions. *The Visual Computer*, in press.
- Hoschek, J. (1971) *Liniengeometrie*. Bibliographisches Institut Zürich.
- Hoschek, J., Lasser, D. (1993) *Fundamentals of Computer Aided Geometric Design*. A K Peters.
- Hoschek, J., Weber, U. (1987) Geometrische Modellbildung der menschlichen Kniebewegung. In: Proceedings of the Congress of Geometry, Thessaloniki, 67–75.
- Hsieh, C.-C., Chang, T.-Y. (2003) Motion fairing using genetic algorithms. *Computer-Aided Design* **35**, 739–749.
- Hyun, D. E., Jüttler, B., Kim, M. S. (2001) Minimizing the distortion of affine spline motions. *Proc. of Pacific Graphics 01*, IEEE Computer Society, 50–59.
- Johnson, D., Cohen, E. (1998) A framework for efficient minimum distance computations. *Proc. International Conference on Robotics and Automation*, Leuven, Belgium, 3678–3684.
- Jüttler, B., Wagner, M. (1996) Computer Aided Design with spatial rational B-spline motions. *ASME J. Mech. Design* **118**, 193–201.
- Jüttler, B., Wagner, M. (2002) Kinematics and Animation. In: Farin, G., Hoschek, J., Kim, M.S., eds., *Handbook of Computer Aided Geometric Design*, Elsevier, 723–748.
- Kobbelt, L. (1996) A variational approach to subdivision. *Computer Aided Geometric Design* **13**, 743–761.
- Kobbelt, L., Schröder, P. (1998) A multiresolution framework for variational subdivision. *ACM Trans. Graphics* **17**, 209–237.
- Latombe, J.C. (2001) *Robot Motion Planning*. 6th printing, Kluwer.
- Marchand, É., Courty, N. (2002) Controlling a camera in a virtual environment. *The Visual Computer* **18**, 1–19.
- Marciniak, K. (1991) *Geometric Modelling for Numerically Controlled Machining*. Oxford University Press, Oxford.
- Park, F.C., Ravani, B. (1995) Bézier curves on Riemannian manifolds and Lie groups with kinematic applications. *ASME J. of Mechanical Design* **117**,

- 36–40.
- Park, F.C., Ravani, B. (1997) Smooth invariant interpolation of rotations. *ACM Transactions on Graphics* **16**, 277–295.
- Pottmann, H., Ravani, B. (2000) Singularities of motions constraint by contacting surfaces. *Mechanism and Machine Theory* **35**, 963–984.
- Pottmann, H., Wallner, J. (2001) *Computational Line Geometry*. Springer-Verlag.
- Röschel, O. (1998) Rational motion design — a survey. *Computer-Aided Design* **30**, 169–178.
- Sharir, M. (1997) Algorithmic motion planning. In: Handbook of Discrete and Computational Geometry. J. E. Goodman and J. O’Rourke, eds., CRC Press, New York, 733–754.
- Shoemake, K. (1985) Animating rotation with quaternion curves. *Computer Graphics* **19**, 245–254.
- Sohn, K. A., Jüttler, B., Kim, M. S., Wang, W. (2002) Computing distances between surfaces using line geometry. *Proceedings of Pacific Graphics 02*, IEEE Computer Society, 236–245.
- Wahba, G. (1990) *Spline Models for Observational Data*, SIAM, Philadelphia.
- Wallner, J. (2002) Gliding spline motions and applications. Technical Report No. 96, Institute of Geometry, Vienna University of Technology.