Optimization-Based Virtual Surface Contact Manipulation at Force Control Rates

Donald D. Nelson Elaine Cohen
University of Utah, Computer Science Dept.
50 S Central Campus Dr. Rm. 3190
Salt Lake City, UT 84112-9205
Email: {dnelson,cohen}@cs.utah.edu

Abstract

Previous interactive works have used springs, heuristics, and dynamics for surface placement applications. We present an analytical technique for kilohertz rate manipulation of CAD models with virtual surface and trimming constraints. The optimization approach allows best placement and sensitivity analysis for mechanical design objectives and parametric domain objectives. Such objectives are not readily incorporated into previous interactive methods. Force feedback is rendered to the user using previously developed haptics principles.

Figure 1: OpenGL^{TM} display and PHANToM^{TM} force feedback interaction with a 3 paper clip surface constraint assembly.

1 Motivation

CAD research into interactive and optimal surface placement is important for prototyping and design variation analysis. Very often, a designer cannot easily determine the best place and orientation for a surface in his virtual still life portrait or CAD assembly. The best surface configuration for some mechanical design objective, such as dexterity, is often desired in CAD prototyping applications. The differential kinematics dexterity measure for example [31],

$$f(\mathbf{q}) = \sqrt{det(\mathbf{J}(\mathbf{q})\mathbf{J}^T(\mathbf{q}))}, \quad (1)$$

where $\mathbf{J}$ is the geometric velocity manipulator Jacobian, can be maximized by an optimization algorithm. It is known that solving high order polynomial objectives must be done through such numerical means. We show how to maximize surface placement objectives, constraints, and constraint Jacobians quickly in this optimization framework.

Furthermore, optimal interactive surface placement supports the generation of physically-based surface animations. A kinematically admissible configuration is required to begin a dynamical simulation. When surfaces are touching, however, a valid initial configuration may be difficult to determine. The interactivity of the optimization environment is important in sketching mechanical configurations at the start of a dynamical simulation.

2 Overview

2.1 Related Work

Surface contact events have been studied extensively for computational dynamics and animation [3, 7, 13, 23]. Interactive surface placement has received attention in the literature through a spring-based collision framework where at any time only a single object may move [34]. A depth-buffer approach has also been used [5]. Optimal surface placement has also been studied with a careful treatment through an expensive (non-interactive) interval analysis [33]. Our method does not cover as broad a class of analysis as the interval methods, but the accompanying global management algorithm allows a very useful class of global optimization to be performed.

The surface-surface floating contact constraint has been evaluated using an expensive finite differences operator [37] and a "distance" Jacobian [1]. In this paper, a fast, closed form parametric contact Jacobian is developed that computes the extremal distance floating contact constraint Jacobian. The parametric contact Jacobian also allows more general types of constraints to be evaluated quickly.

Algorithms to solve the geometric satisfaction problem [11], inverse kinematics problem [20, 36], and nonlinear constrained optimization problems have been developed using graph theory and numerical programming means. Mechanisms that self-assemble through inverse dynamics "forces" for simple joint constraints have been shown to be effective in modeling applications [4]. Differential manipulation for virtual peg in hole and other basic constraints has been studied but without emphasis on design objectives or force feedback [12]. The methods used in our paper will focus on kinematic optimization to provides positions, velocities, and accelerations over time to our inverse dynamics controller.

2.2 Representation

The interactive rate solution presented relies on fast NURBS surface evaluation (i.e. computation of Eq. 2 and its derivatives) since the parametric contact Jacobians require two surface evaluations. An optimized version of the Alpha matrix refinement operations [9] in our CAD system has been implemented to achieve a rate of 30 kilohertz for surface evaluations.

The piecewise tensor product non-uniform rational B-spline definition of a surface $\mathbf{S}$ is a mapping from $R^2 \rightarrow R^3$, i.e. a function from parametric (u,v) space to Cartesian (x,y,z) space.

$$\overline{\mathbf{S}}(u,v) = \frac{\sum_{i,j} w_{i,j}\overline{\mathbf{p}}_{i,j}B_{i,k_u}(u)B_{j,k_v}(v)}{\sum_{i,j} w_{i,j}B_{i,k_u}(u)B_{j,k_v}(v)} = \begin{bmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{bmatrix} \quad (2)$$

where the B-spline blending functions B, control mesh $\overline{\mathbf{p}}$, and weights $\mathbf{w}$ are used. The over-line notation such as $\overline{\mathbf{S}}$ is used to indicate the local body frame.

Use of minimal coordinate representations such as Denavit-Hartenburg or Hayati parameters, would need to be extended to include surface interactions. We use the augmented generalized coordinate representation of mechanical systems [14, 32] to incorporate both simple mechanical joints and complex surface constraints. The position and quaternion orientation coordinates for body i are denoted by $\mathbf{q}^i$. The parametric contact coordinates $\mathbf{u} = [u^i\ v^i\ u^j\ v^j]^T$ for two surfaces $\mathbf{i}$ and $\mathbf{j}$ in contact are completely dependent on $\mathbf{q}$ and are therefore removed from our coordinates to be optimized.

3 Fast Analysis of Surface Constraints

Our approach to high speed, interactive surface configuration optimization leverages constraints and constraint Jacobians to define a kinematically admissible configuration space and admissible motion. This information drives the high speed optimization algorithm at force control rates.

3.1 Design Objectives

We describe both simple and complex mechanical joints as geometric constraints. Since the system is under-specified, there are infinitely many solutions that satisfy these constraints for mechanisms that move, or have degrees of freedom (DOFs). Mechanical design objectives are introduced to specify the best solution choice. For virtual manipulation of the closed loop mechanism depicted at the end of this work (Fig. 7), the design objective is to minimize the distance from the middle link to the hand grasp. The grasp can be modeled as point contact, rigid hand contact, and the surface-surface floating contact described here. This design objective extends previous work in virtual inverse kinematics manipulation [12, 38, 27]. Other mechanical design measures with good convergence properties that have been implemented to further constrain the system are dexterity eigenvector ellipsoids, workspace volume, and force transmission measures.

3.2 Extremal Distance Computations

The bilateral surface contact constraint which ensures that two surfaces $\mathbf{f}$ and $\mathbf{g}$ touch is given by

$$min_{\mathbf{u}^f,\mathbf{g}^f} ||\mathbf{f}(\mathbf{u}^f) - \mathbf{g}(\mathbf{u}^g)|| = 0, \quad (3)$$

which removes one DOF from the solution manifold. This constraint is to be maintained regardless of the current model configuration. The zero distance constraint in this case is called extremal distance, as opposed to minimal distance, because $\mathbf{f}$ and $\mathbf{g}$ may be interpenetrating. The extremal distance contact points have been studied with gradient descent [34] and improved Newton iterative distance approaches [18, 28]. The velocity formulation of Eq. 4 guarantees the maintenance of the extremal distance solution since it is an exact computation, not an iterative numerical method, and cannot not slide into extraneous, undesirable solutions. If there is an extremal distance solution, the velocity method will compute it, but there are unusual cases where infinitely many extremal solutions exist. The current solution is acceptable in the case of infinite solutions.

3.3 Surface Constraint Jacobian Extraction

The Jacobian relating the extremal distance contact coordinates with the linear and quaternion velocities of the surfaces in contact can be written in the form (as derived in Appendix A)

$$\begin{bmatrix} \dot{\mathbf{u}}^f \\ \dot{\mathbf{u}}^g \end{bmatrix} = \mathbf{AB} \begin{bmatrix} \dot{\mathbf{q}}^f \\ \dot{\mathbf{q}}^g \end{bmatrix}, \quad (4)$$

where $\mathbf{AB}$ is the Jacobian $\mathbf{u_q}$ relating differentials $\dot{\mathbf{u}}$ and $\dot{\mathbf{q}}$.[1]

Since closed forms for the quantities $\mathbf{S_p}$, $\mathbf{p_q}$ (see Appendix B), and $\mathbf{S_u}$ are easily derived, the solution is given by $\mathbf{S_q} = \mathbf{S_u u_q} + \mathbf{S_{up} p_q}$, are also conveniently obtained once $\mathbf{u_q}$ is known. $\mathbf{u_q}$ and $\mathbf{S_q}$ will be used in the kilohertz update optimization algorithm.

The Jacobian of the constraint that two surfaces must touch anywhere (are floating) is readily obtained as the Jacobian of Eq. 3 since $\mathbf{f_q}$ and $\mathbf{g_q}$ are available from Eq. 4. I.e., $||\mathbf{f}(\mathbf{u}^f) - \mathbf{g}(\mathbf{u}^g)||_{\mathbf{q}}|| = \beta_{\mathbf{q}} = \beta^{-1}(\mathbf{f}-\mathbf{g})^T(\mathbf{f_q}-\mathbf{g_q})$. The situation becomes more interesting for the derivation of the Jacobians of parametric contact constraints, used in parametric trimming loop constraints.

3.4 Trimmed Surface Constraints

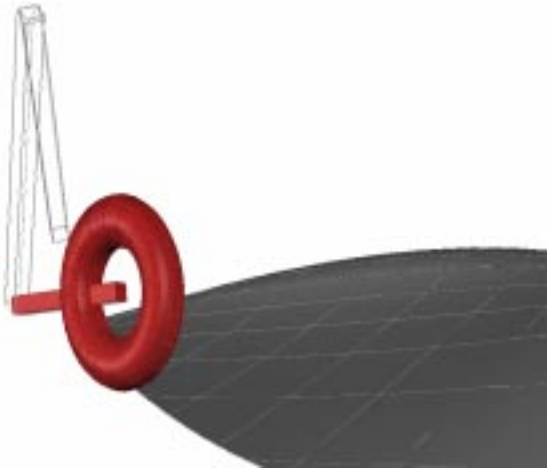

Figure 2: An additional condition is included to avoid placement in trimmed away areas, such as the boundary of the region on the right. The parametric constraint Jacobian for trimming constraints is obtained in closed form through the surface-curve velocity relations.

Trimming curves occur in CAD modeling when boolean set or sculpting operations are used. A curve $\mathbf{c}$ embedded in a surface $\mathbf{S}$ refers to the composition $\mathbf{S}(\mathbf{c})$. $\mathbf{c}(t)$ is a planar curve mapping a scalar to R^2, the (u,v) parametric domain of the surface. If $\mathbf{f}$ or $\mathbf{g}$ has a curve embedded in it (Fig. 2), such as a trimming curve, we can write the constraint Jacobian of the following curve-surface constraint since we can extract the Jacobian from the curve-surface velocity method in Appendix A. Other types of parametric curve constraints can also be written in a similar manner.

Figure 4 illustrates the unsatisfied, initial configuration of an assembly with surface-surface contact with trims. There are several options for writing the surface-curve unilateral or bilateral constraint for placement in a parametric area or avoiding trimming loops. We have identified and implemented three versions:

[1] The subscript $\mathbf{q}$, $\mathbf{u}$ and $\mathbf{p}$ will denote the partial derivative with respect to a vector, i.e. the Jacobian.

Figure 3: Global considerations are managed through polygonal and numerical tracking methods. Two surface contact constraints are shown in this paperclip example. The global method coordinates with the local, analytic method to add and remove global collision constraints.

1. Parametric domain: $||\mathbf{u}^f_{extrema} - \mathbf{u}^f_{curve}|| = 0$

2. Cartesian domain: $||f(\mathbf{u}^f_{extrema}) - f(\mathbf{u}^f_{curve})|| = 0$

3. Cartesian domain between surfaces: $||g(\mathbf{u}^g_{extrema}) - f(\mathbf{u}^f_{curve})|| = 0$,

where the subscript $extrema$ refers to the extremal distance parametric coordinate between surfaces and the subscript $curve$ denotes the extremal distance coordinate of the curve one embedded on one surface with respect to the other surface. The Jacobian of those constraints is easily derived since we have $\mathbf{u}^f_{curve,\mathbf{q}}$, $\mathbf{f}(\mathbf{u}^f_{curve})_{\mathbf{q}}$, $\mathbf{u}^f_{extrema,\mathbf{q}}$, and other derivatives.

4 Constraint Satisfaction

An augmented Lagrangian approach [10, 29, 30] has been implemented to solve the constrained optimization problem. The solution to any nonlinear constrained optimization problem is expensive in general, but our problem domain takes advantage of the coherence in the solution between optimization iterations. The convergence properties are discussed in the results section.

Surface constraints may become inactive during the course of the optimization or manipulation of the virtual surface assembly. The detection and tracking of these events is performed in a global management algorithm. When a contact event has occur, the corresponding constraint is added or deleted.

5 Global Collision Management

The Jacobian extraction in Eq. 4 relies on the contact point being maintained exactly through a combined local and global tracking process. We have implemented a three step tracking process to manage the global issues that are not guaranteed by the local collision methods. The details were reported in previously as part of results on a haptics surface-surface tracing algorithm [28]. Our

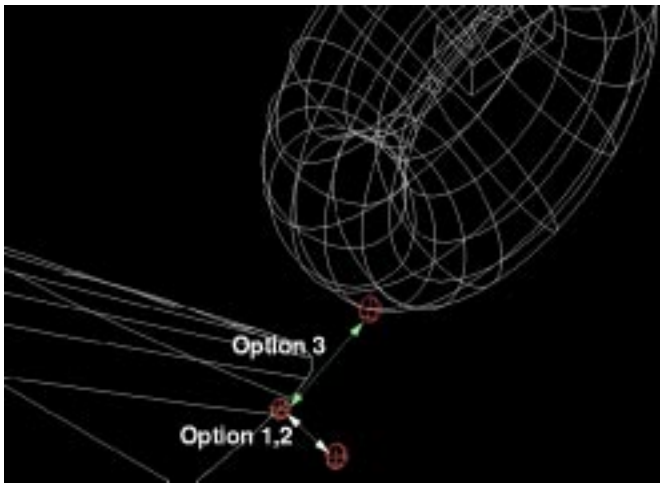

Figure 4: Unsatisfied surface-surface contact constraint showing minimal distance between the surfaces and surface-curve minimal distance. Several constraint domains to write the surface-curve inequality condition are possible.

manager uses a global polygonal collision algorithm [21] on an offset model to start a Newton tracking process for exact tracking of locally closest points. These parametric tracking coordinates are input to the Jacobian extraction equations.

The "spheres-of-neglect" check [34] has been added to avoid artifacts caused by the polygonal representation for global collisions. More than 1 polygon pair may intersect where there is only a single contact point in the smooth representation. However, Figure 3 illustrates a situation where a very thin-radius paperclip may have 2 or more smooth surface contact points within a small "radius". The problem is hard in general, requiring an expensive, exact global intersection algorithm. We have added a check to make sure that the neglect does not exclude areas that are not nearby topologically, as in the outer and inner loop of the paperclip. The class of surfaces that do not have tiny, undulating features (convex surface areas with a radius of curvature smaller than that of the neglect radius) are handled well with the added second check.

6 Inverse Dynamics Feedback

Inertial and constraint force feedback also provide important prototyping information. Constraint forces and inverse dynamics forces are computed according to methods for virtual mechanical assemblies previously developed by the authors [29]. Force in configuration space of the mechanism $\mathbf{W}_c$, such as in the Cartesian/quaternion set of coordinates, may be expressed in terms of body space force and torque $\mathbf{W}$ through standard transformation operators, denoted by $\mathbf{G}$ [32, 14],

$$\mathbf{W} = \frac{1}{4}\mathbf{G}(\mathbf{q})\mathbf{W}_c \tag{5}$$

This force and torque may be projected onto joint axes to provide the amount of torque required by a controls system to produce a motion. For our interactive haptics force feedback application, $\mathbf{W}$ can be transmitted to the user.

7 3 and 6 Axes of Force Feedback

Our system drives a PHANToM^{TM} haptics device that senses 6 DOFs of position and renders 3 DOFs of force. The generalized

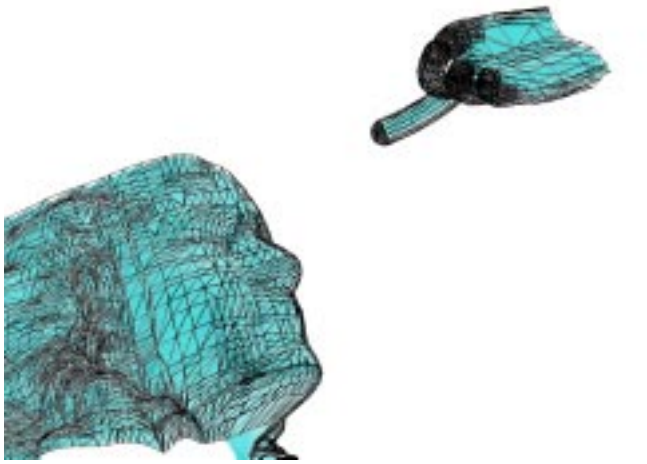

Figure 5: The global polygonal manager is used in the "far" regions on an offset of the model surface. This result starts a nonlinear root finger for tracking "near" cases. Each root is used and updated by the velocity formulation during the local tracking case.

wrench to be rendered to the user, $\mathbf{W}$, is input into the algorithm. The first 3 components of $\mathbf{W}$ are taken as output for the 3 DOF PHANToM case.

For a more complicated anthropomorphic haptic mechanism providing feedback to a user's entire arm, we use a relation from statics to project the virtual wrench to the appropriate actuators [35]. The following equations allow the use of a tool frame for the finger without changing the manipulator Jacobian.

$$\mathbf{W} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{r}_1 \times \mathbf{f}_1 \end{bmatrix}, \tag{6}$$

$$\mathbf{J} = \begin{bmatrix} \mathbf{z}_0 \times \mathbf{b}_0 & \dots & \mathbf{z}_4 \times \mathbf{b}_4 & 0 & 0 & 0 \\ \mathbf{z}_0 & \dots & \mathbf{z}_4 & \mathbf{z}_5 & \mathbf{z}_6 & \mathbf{z}_7 \end{bmatrix}, \tag{7}$$

$$\tau = \mathbf{J}^T \mathbf{W}, \tag{8}$$

where $\mathbf{W}$ is the wrench at the wrist, f_1 is the force at the finger tip, r_1 is the moment arm of the finger on the wrist, z_i is the axis of rotation for joint i, $\mathbf{b}_i$ is a vector from the origin of joint i to the wrist, τ is the haptics control torque vector, and $\mathbf{J}$ is the manipulator Jacobian. The force reflecting arm can be an exoskeleton worn by the user. In this example, the shoulder is spherical attachment from revolute joints 1,2,3. The elbow is joint 4. The wrist is spherical, composed of revolute joints 5, 6, 7.

8 Results

This approach has been incorporated into the Alpha_1 CAD system [9].

8.1 Iteration Rates

The size of the control mesh does not slow down the local constraint analysis since the surface evaluations are done with Alpha_1 refinement operations. The global constraint tracking depends on the complexity of the surfaces being analyzed, but a hierarchical, polygonal global management scheme is employed that reduces this cost so that it is not the bottleneck in the parallel management implementation.

4 kHz updates are achieved for the 3 surface paperclip assembly on a four processor SGI 300MHz R12000. The method is scalable to any number of surface contact constraints since the surfaces can be evaluated in parallel within an optimization iteration. There is no parallelism between iterations. For the examples used to date, the surface evaluations, not the optimization computation, are the dominant cost. Highly parallel optimization algorithms are a subject of future work.

8.2 Convergence Rates

A number of measures have been introduced in this work to allow multiple, highly nonlinear surface and trimming constraints to be satisfied while maximizing mechanical design objectives. This convergence is an intractable problem in general (note that it is easy to write a system of constraints that cannot be satisfied). However, our monitoring mechanism simplifies the problem by keeping track of global collision events. This simplification allows the local extremal distance contact computations to be maintained exactly, without problems that numerical methods suffer. The collision management scheme also allows the curve extremal distance to be maintained exactly. For surfaces that do not have tiny, sharp changes as described in our section on "spheres-of-neglect," the proper extremal distance can be maintained.

For the steady state case (during PHANToM manipulation), the second order gradient, or Hessian, can be computed to check whether it is positive definite. In practice, the user cannot move so quickly as to escape the region where the solution space is convex (positive definite Hessian); therefore, the system will converge. The norm of the constraint manifold, a constraint violation measure, is less than 10^{-5} for simple mechanisms on the scale of 1 unit in size.

9 Research Conclusions

We have established a principled method to handle force-interactive surface placement for a large class of mechanical design objectives. This is a vast improvement over .1-20 Hz methods (adjusted for current machine speeds) used in previous works. Mechanical design objectives such as placement for maximal dexterity are readily encoded in an optimization framework. The addition of objectives to prescribe unconstrained degrees of freedom is a very useful and important separation of this method with previous manipulation algorithms. The ability to analyze the surface constraints at force update rates is the critical problem solved in our work.

Appendix A: Surface Contact Velocity Formulation

A number of different derivations of the kinematics of contact have been developed in the last 15 years. These formulations relate the rate of change of the parametric contact coordinates to the Cartesian velocity and angular velocity of the bodies in contact. Previous works have been limited to surfaces parameterized to have orthogonal surface partial derivatives. Some are also limited the in-contact case[25, 26, 6]. While any surface may be reparameterized to be orthogonal, it may be expensive and impractical to find such a parameterization. The finger may bend in our interactive application. Finding the reparameterization with full numerical precision is also a problem[22]. We develop a new derivation for the not-in-contact, non-orthogonal surface parameterizations.

When both surfaces have partials that are everywhere orthogonal, i.e. $\mathbf{f}_u \cdot \mathbf{f}_v = 0$ and $\mathbf{g}_s \cdot \mathbf{g}_t = 0$, $\forall \mathbf{u}$ in the parametric domain, an original result from [25], extended by [1] for the not in contact case, derives the surface kinematics as

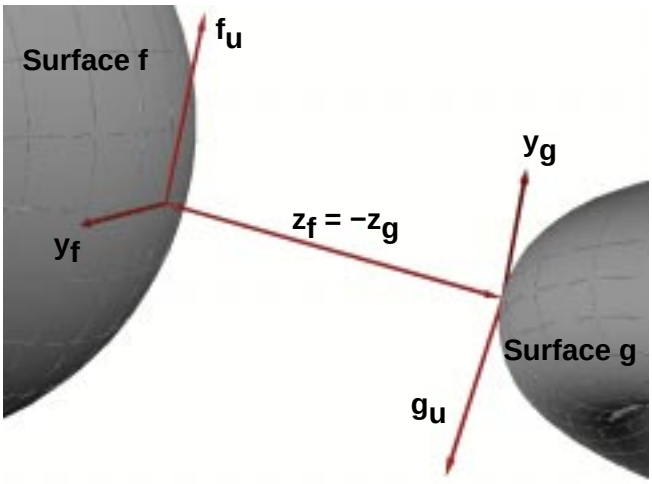

Figure 6: Closest point surface contact frames. Velocity relations allow the contact coordinate velocities to be found.

$$\dot{\mathbf{u}}^f = {\mathbf{I}_f}^{-1}\mathbf{R}_\theta(\mathbf{II}_g + \widetilde{\mathbf{II}}_f + \beta\mathbf{II}_g\widetilde{\mathbf{II}}_f)^{-1}(\begin{bmatrix} -w_y \\ w_x \end{bmatrix} + \mathbf{II}_g \begin{bmatrix} v_x \\ v_y \end{bmatrix})$$

$$\dot{\mathbf{u}}^g = {\mathbf{I}_g}^{-1}(\mathbf{II}_g + \widetilde{\mathbf{II}}_f + \beta\widetilde{\mathbf{II}}_f\mathbf{II}_g)^{-1}((\mathbf{1} + \beta\widetilde{\mathbf{II}}_f)\begin{bmatrix} -w_y \\ w_x \end{bmatrix} - \widetilde{\mathbf{II}}_f \begin{bmatrix} v_x \\ v_y \end{bmatrix})$$

where β is the (signed) distance between contacts, $\mathbf{1}$ is the 2×2 identity matrix, the relative linear and angular surface velocities of surface $\mathbf{g}$ relative to surface $\mathbf{f}$ be denoted by $\mathbf{v}, \omega$, the surface contact velocities be $\mathbf{v}^f, \omega^f$ and $\mathbf{v}^g, \omega^g$, $\mathbf{I}$ is the first fundamental form and $\mathbf{II}$ is the surface curvature or second fundamental form, with subscripts for surfaces $\mathbf{f}$ and $\mathbf{g}$. θ represents the angle between parametric axes $\mathbf{g}_s$ and $\mathbf{f}_u$, let $\mathbf{R}_\theta = \begin{bmatrix} cos\theta & -sin\theta \\ -sin\theta & -cos\theta \end{bmatrix}$ and $\widetilde{\mathbf{II}} = \mathbf{R}_\theta\mathbf{II}\mathbf{R}_\theta$. The relation may be written in the following matrix form,

$$\dot{\mathbf{u}} = \mathbf{A}\begin{bmatrix} \mathbf{v}_{x,y} \\ \omega_{x,y} \end{bmatrix}. \tag{9}$$

Non-Orthogonal Parameterizations

We provide a new derivation for $\mathbf{A}$ for regular parametric surfaces whose partials are not orthogonal so the result in Eqn. 9 can be used for typical models constructed by CAD systems.

The parametric contact frames in Fig. 6 for the extremal distance context are defined with an orthonormal set of vectors. $\mathbf{R}_f = [\mathbf{x}_f \quad \mathbf{y}_f \quad \mathbf{z}_f]$ is the rotation matrix from the local contact frame to the world frame, where $\mathbf{x}_f = \mathbf{f}_u/||\mathbf{f}_u||$, $\mathbf{z}_f = \mathbf{f}_u \times \mathbf{f}_v/||\mathbf{f}_u \times \mathbf{f}_v||$, $\mathbf{y}_f = \mathbf{z}_f \times \mathbf{x}_f$. $\mathbf{R}_g$ is similarly defined. $\mathbf{z}_f$ and $\mathbf{z}_g$ are parallel free vectors (Fig. 6).

Comparison of the relative surface velocities[2] can be used to relate the parametric contact coordinates $\mathbf{u}^f$ and $\mathbf{u}^g$ with the linear and angular surface velocities. Let the surface velocities $\mathbf{v}^f$ and $\mathbf{v}^g$ and relative surface velocity $\mathbf{v}$ be in the frame $\begin{bmatrix} \mathbf{x}_g & \mathbf{y}_g & \mathbf{z}_g \end{bmatrix}$. In the extremal distance context (see Fig. 6), we have

$$\mathbf{v}^g_{x,y} + \mathbf{v}_{x,y} = \mathbf{R}_\theta\mathbf{v}^f_{x,y} + \beta\mathbf{R}_\theta\omega^f_{y,-x}, \tag{10}$$

$$\omega^g_{y,-x} + \omega_{y,-x} = -\mathbf{R}_\theta\omega^f_{y,-x}. \tag{11}$$

The terms $\mathbf{v}^f$ and $\mathbf{v}^g$ will contain $\dot{\mathbf{u}}^f$ and $\dot{\mathbf{u}}^g$ in the following equations through due to the chain rule of differentiation. We derive matrices $\mathbf{E}^f_{x,y}$ and $\mathbf{F}^f_{x,y}$ for surface f (and analogously for surface g) from the relations of linear and angular velocity to separate $\dot{\mathbf{u}}^f$ and $\dot{\mathbf{u}}^g$ from other terms,

$$\mathbf{v}^f_{x,y} = \mathbf{R}^T_{f_{x,y}}\dot{\mathbf{f}}_{x,y} = \left[\begin{bmatrix} \mathbf{x}_f & \mathbf{y}_f \end{bmatrix}^T \mathbf{f_u}\right]_{2\times 2}\dot{\mathbf{u}}_f = \mathbf{E}^f_{x,y}\dot{\mathbf{u}}^f, \tag{12}$$

$$\omega^f_{y,-x} = extract\_skew\_symmetric({\mathbf{R}_f}^T\dot{\mathbf{R}}_f)_{y,-x}$$

$$= \begin{bmatrix} {\mathbf{x}_f}^T\mathbf{z_u} \\ {\mathbf{y}_f}^T\mathbf{z_u} \end{bmatrix}_{2\times 2}\dot{\mathbf{u}}^f = \mathbf{F}^f_{y,-x}\dot{\mathbf{u}}^f. \tag{13}$$

Using Eqns. 10,11,12,13, the general non-orthogonal case is reduced to the 4×4 system,

$$\begin{bmatrix} \mathbf{R}_\theta(\mathbf{E}^f_{x,y} - \beta\mathbf{F}^f_{-y,x}) & -\mathbf{E}^g_{x,y} \\ (-\mathbf{R}_\theta\mathbf{F}^f_{y,-x})_{x,y} & -\mathbf{F}^g_{x,y} \end{bmatrix}\begin{bmatrix} \dot{\mathbf{u}}^f \\ \dot{\mathbf{u}}^g \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{x,y} \\ \omega_{x,y} \end{bmatrix}. \tag{14}$$

This system can be solved quickly for $\dot{\mathbf{u}}$ using the following pseudocode fragment for arbitrary surface parameterizations. We rewrite the inverse of the 4×4 coefficient matrix in Eq. 14,

$$\mathbf{A} = \begin{bmatrix} \mathbf{R}_\theta(\mathbf{E}^f_{x,y} - \beta\mathbf{F}^f_{-y,x}) & -\mathbf{E}^g_{x,y} \\ -(\mathbf{R}_\theta\mathbf{F}^f_{y,-x})_{x,y} & -\mathbf{F}^g_{x,y} \end{bmatrix}^{-1} \tag{15}$$

to be solved even more efficiently as a series of 2×2 matrix inverses and multiplications,

```
iEg_xy=inv(Eg_xy);
RphiFf_xy=Rphi*Ff_xy;
dRphiFf_xy=d*RphiFf_xy;
Fg_xyiEg_xy=Fg_xy*iEg_xy;
RphiEf_xy=Rphi*Ef_xy;

H=Fg_xyiEg_xy*(RphiEf_xy+dRphiFf_xy)-
RphiFf_xy;
iH=inv(H);
J=iH*[Fg_xyiEg_xy -eye(2)];

A=[J; iEg_xy*(RphiEf_xy*J+
        dRphiFf_xy*J-[eye(2) zeros(2)])];
```

The solution is roughly as efficient as previous methods since the inverse of two 2×2 matrices and nine matrix multiplications, rather than four 2×2 matrix inversions and six matrix multiplications, is required (see pseudocode fragment). The optimized matrix inverse and multiplication implementation is a constant cost and is a small fraction of the cost associated with a surface evaluation.

[2] A subscript with x or y such as $\mathbf{a}_x$ will denote the first component of the vector. Several subscripts, such as $\mathbf{a}_{-x,y}$ will represent a two-vector containing the negative of the first component and the second component of $\mathbf{a}$. A superscript T as in $\mathbf{a}^T$ will denote a vector or matrix transpose. Partitions of matrices may be selected with (row,column) indexing, with "a:b" for a range or ":" for all rows or columns, as in the $Matlab^{TM}$ notation. The operator $extract\_skew\_symmetric$ retrieves 3 independent components from 9 elements of a skew symmetric matrix.

Cartesian and Quaternion Generalized Velocities

Now we express the relative surface velocities in terms of world frame body coordinate velocities so that integration of orientation is possible (integration of angular velocity is meaningless). Define the local contact frame through the rotation matrix

$$\mathbf{R}_{loc} = \left[\begin{array}{ccc} \mathbf{x}_g & \mathbf{y}_g & \mathbf{z}_g \end{array} \right]^T. \quad (16)$$

The elements of a quaternion $\mathbf{q}_{rot}$ will be denoted by q_{rot_0}, q_{rot_x}, q_{rot_y}, q_{rot_z}. Let $\mathbf{G}_f = \mathbf{G}(\mathbf{q}_{f,rot})$, $\mathbf{G}_g = \mathbf{G}(\mathbf{q}_{g,rot})$, where $\mathbf{G}(\mathbf{q}_{rot})$ is the matrix operator mapping quaternion velocities to angular velocities [14, 32], given by

$$\mathbf{G}(\mathbf{q}_{rot}) = 2 \left[\begin{array}{cccc} -q_{rot_x} & q_{rot_0} & q_{rot_z} & -q_{rot_y} \\ -q_{rot_y} & -q_{rot_z} & q_{rot_0} & q_{rot_x} \\ -q_{rot_z} & q_{rot_y} & -q_{rot_x} & q_{rot_0} \end{array} \right]. \quad (17)$$

The velocity $[\mathbf{v}^T \omega^T]^T$ is the motion of surface g relative to surface f. We write our world space velocities in terms of the local frame. We relate relative angular velocity and world frame quaternion velocity by

$$\omega = \mathbf{R}_{loc} \left[\begin{array}{cc} -\mathbf{T}_f \mathbf{G}_f & \mathbf{T}_g \mathbf{G}_g \end{array} \right] \left[\begin{array}{c} \dot{\mathbf{q}}_{f,rot} \\ \dot{\mathbf{q}}_{g,rot} \end{array} \right], \quad (18)$$

and relative linear velocity to Cartesian velocity through

$$\mathbf{v} = \mathbf{R}_{loc}(\dot{\mathbf{q}}_{g,tr} - \dot{\mathbf{q}}_{f,tr} + (\omega_g^{gl} - \omega_f^{gl}) \times (\mathbf{g}(\mathbf{u}) - \mathbf{q}_{g,tr})). \quad (19)$$

$\times$ denotes the vector cross product. $\mathbf{T}_f$ and $\mathbf{T}_g$ are rotations from the local frame to the world frame defined by quaternions $\mathbf{q}_{f,rot}$ and $\mathbf{q}_{g,rot}$. It can be shown that the relative surface velocities $[\mathbf{v}^T, \omega^T]^T$ are related to Cartesian and quaternion velocities through

$$\left[\begin{array}{c} \mathbf{v} \\ \omega \end{array} \right] = \left[\begin{array}{cccc} -\mathbf{R}_{loc} & \mathbf{0}_{3\times 4} & \mathbf{R}_{loc} & -\mathbf{R}_{loc}((\mathbf{g}(\mathbf{u}) - \mathbf{q}_{g,tr})\times)\mathbf{R}_g \mathbf{G}_g \\ \mathbf{0}_{3\times 7} & & & \mathbf{R}_{loc}\mathbf{R}_g \mathbf{G}_g \end{array} \right] *$$

$$\left[\begin{array}{cccc} \mathbf{I}_{3x3} & \mathbf{0}_{3x11} & & \\ \mathbf{0}_{4x14} & & & \\ \mathbf{0}_{3x3} & ((\mathbf{g}(\mathbf{u}) - \mathbf{q}_{g,tr})\times)\mathbf{R}_f \mathbf{G}_f & \mathbf{I}_{3x3} & \mathbf{0}_{3x4} \\ \mathbf{0}_{4x3} & -\mathbf{G}_g^T \mathbf{R}_g^T \mathbf{R}_f \mathbf{G}_f & \mathbf{0}_{4x3} & -\mathbf{G}_g^T \mathbf{G}_g \end{array} \right] \left[\begin{array}{c} \dot{\mathbf{q}}^f \\ \dot{\mathbf{q}}^g \end{array} \right] \quad (20)$$

where $\times$ in Eq. 20 denotes the $3x3$ skew symmetric matrix that performs the operation of a cross product (obtained from the three components of a vector, i.e. $\mathbf{a}\times = \left[\begin{array}{ccc} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{array} \right]$). From Eq. 14, the truncated part $[\mathbf{v}_{x,y}^T, \omega_{x,y}^T]^T$ is all that is required. Let $\mathbf{B}$ contain the first two rows and rows four and five of Eq. 20. Substituting $[\mathbf{v}_{x,y}^T, \omega_{x,y}^T]^T$ into Eq. 14, yields

$$\dot{\mathbf{u}} = \mathbf{A}\mathbf{B} \left[\begin{array}{c} \dot{\mathbf{q}}^f \\ \dot{\mathbf{q}}^g \end{array} \right]. \quad (21)$$

Non-Orthogonal Surface-Curve Velocity Formulation

Similarly, it may be shown that the surface-curve extremal distance equations may be extended to for arbitrary surface parameterizations.

The time derivative of the parametric contact coordinates for surface f and curve g may be written as a function of linear and angular velocity multiplying a matrix operator,

$$\dot{\mathbf{u}}_{3x1} = \left[\begin{array}{cc} \mathbf{R}_\theta(\mathbf{E}_{\mathbf{u}}^f + d\mathbf{F}_{x,y}^f) & -\mathbf{g}_{u\,x,y} \\ -\mathbf{R}_\theta \mathbf{F}_{x,y}^f & -k_g sin(\phi)\mathbf{g}_{u\,x,y} \end{array} \right]_{3x3}^{-1} \left[\begin{array}{c} \mathbf{v}_{x,y} \\ \omega_y \end{array} \right]_{3x1}, \quad (22)$$

where

$$k_g = \frac{||\mathbf{g}_u \times \mathbf{g}_{uu}||}{||\mathbf{g}_u||^3} \quad (23)$$

and ϕ is the angle between the curve normal (which is $-\mathbf{z}^f$ for the extremal distance case) and the curve binormal $\mathbf{b}$, about the curve x axis $\mathbf{g}_u$. The binormal is

$$\mathbf{b} = \frac{\mathbf{g}_u \times \mathbf{g}_{uu}}{||\mathbf{g}_u \times \mathbf{g}_{uu}||}. \quad (24)$$

Eq. 20 is again used to establish this relation in terms of quaternion and Cartesian velocities, using rows 1,2, and 5.

Appendix B: Quaternion Differential Algebra

$\mathbf{q}_{rot}$ rotates a constant (i.e. local frame) vector $\mathbf{v}$ using quaternion multiplication, given by:

$$\mathbf{q}_{rot}\mathbf{v}\mathbf{q}_{rot}^* = (q_0^2 - \mathbf{q}_{rot} \cdot \mathbf{q}_{rot})\mathbf{v} + 2q_0 \mathbf{q}_{rot} \times \mathbf{v} + 2\mathbf{q}_{rot}(\mathbf{q}_{rot} \cdot \mathbf{v})$$

where $\mathbf{q}_{rot}^* = q_0 - \mathbf{q}_{rot}$ denotes the inverse of the quaternion.

The important Jacobian operation of the expression $\mathbf{q}_{rot}\mathbf{v}\mathbf{q}_{rot}^*$ with respect to the rotation $\mathbf{q}_{rot}$ used in simple joints and more complicated surface constraints is now given by

$$\frac{\partial(\mathbf{q}_{rot}\mathbf{v}\mathbf{q}_{rot}^*)}{\partial \mathbf{q}_{rot}} = \left[\begin{array}{cc} q_y v_z - q_z v_y & q_x v_x + q_y v_y + q_z v_z \\ q_0 v_y + q_z v_x - q_x v_z & -q_x v_y - q_0 v_z + q_y v_x \\ q_0 v_z + q_x v_y - q_y v_x & q_0 v_y + q_z v_x - q_x v_z \end{array} \right.$$

$$\left. \begin{array}{cc} q_0 v_z + q_x v_y - q_y v_x & -q_z v_x - q_0 v_y + q_x v_z \\ q_x v_x + q_y v_y + q_z v_z & -q_z v_y + q_0 v_x + q_y v_z \\ -q_y v_z - q_0 v_x + q_z v_y & q_x v_x + q_y v_y + q_z v_z \end{array} \right] \quad (25)$$

References

[1] Anitescu, M., Cremer, J., and Potra, F., "Formulating 3D Contact Dynamics Problems," *Mech. Struct. & Mach.*, vol. 24, no. 4, pp. 405-437, Nov. 1996.

[2] Ashrafiuon, H., Mani, N., "Analysis and Optimal Design of Spatial Mechanical Systems," in Journal of Mechanical Design, Transactions of the ASME, Vol. 112, No 2., June 1990.

[3] Baraff, D., "Curved surfaces and coherence for non-penetrating rigid body simulation", in *Computer Graphics Proceedings, SIGGRAPH*, 24(4): 19-28, 1990.

[4] Barzel, R., Barr, A., "A Modeling System Based On Dynamic Constraints," in *Computer Graphics*, pp. 179-187, ACM, August 1988.

[5] Breen, David E., Whitaker, R., Rose, E., and Tuceryan, M., "Interactive occlusion and automatic object placement for augmented reality," Eurographics '96 Proceedings, August 1996, pp. 11-22.

[6] Cai, C., Roth, B., "On the spatial motion of rigid bodies with point contact," in *Int. Conf. Robotics and Automation*, pp. 686-695, March 1987.

[7] Carpenter, N., Taylor, R., Katona, M., "Lagrange constraints for transient finite element surface contact," in *International Journal for Numerical Methods in Engineering*, vol. 32, no. 1, July, 1991, pp. 103-128.

[8] Cheney, W., Kincaid, D., *Numerical Mathematics and Computing*, Brooks/Cole Publishing, 1985.

[9] Cohen, E., Lyche, T., and Riesenfeld, R., "Discrete B-Splines And Subdivision Techniques In Computer Aided Geometric Design And Computer Graphics," *Computer Graphics and Image Processing*, Vol 14, Number 2, October 1980.

[10] Choi, M., Cremer, F., "Geometric Awareness for Interactive Object Manipulation," in *Proceedings of Graphics Interface*, 1999.

[11] Fudos, I., Hoffmann, C., "A Graph-Constructive Approach to Solving Systems of Geometric Constraints," in *ACM Transactions on Graphics*, Vol. 16, No. 2, April 1997, pp. 179-216.

[12] Gleicher, M., A Differential Approach to Graphical Interaction. Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, 1994.

[13] Goyal, S., "Second Order Kinematic Constraint Between Two Bodies Rolling, Twisting and Slipping Against Each Other While Maintaining Point Contact," Technical Report, Dept. Comp. Sci, Cornell University, 1989.

[14] Haug, E., Intermediate Dynamics, Prentice Hall, 1992.

[15] Johnson, A., "An Open Architecture Approach to Kinematic Analysis for Computer-Aided Embodiment Design," in *Computer Aided Design*, Vol 30. no. 3, March 1998, pp. 169-178.

[16] Johnson, D.E., and Cohen, E., "An improved method for haptic tracing of sculptured surfaces," Symp. on Haptic Interfaces, ASME International Mechanical Engineering Congress and Exposition, Anaheim, CA, Nov. 15-20, 1998, in press.

[17] Kramer, G. A., Solving geometric constraint systems : a case study in kinematics, McGraw Hill, 1992.

[18] Kriezis, G., Patrikalakis, N., Wolder, F., "Topological and differential-equation methods for surface intersections," in *Computer-Aided Design*, vol. 24, no. 1, January 1992, pp. 41-51.

[19] Li, Z., Canny, J., "Motion of Two Rigid Bodies with Rolling Contact," in *IEEE Trans. on Rob. and Autom.*, vol. 6, no. 1, 1990.

[20] Lenarcic, J., "Alternative Computational Scheme of Manipulator Inverse Kinematics", in *International Conference on Robotics and Automation*, pp. 3235-3240. 1998.

[21] M.C. Lin, D. Manocha, and J. Canny. Fast contact determination in dynamic environments. in *IEEE Conference on Robotics and Automation*, pp. 602-609, 1994.

[22] Maekawa, T., Wolter, F., Patrikalakis, N., "Umbilics and lines of curvature for shape interrogation," in *Computer Aided Geometric Design*, vol. 13, no. 2, March 1996.

[23] Marigo, A., Bicchi, A., "Rolling bodies with regular surface: the holonomic case," Differential Geometry and Control, Guillermo Ferreyra, Robert Gardner, Henry Hermes and Hector Sussmann (eds.), Proceedings of Symposia in Pure Mathematics, American Mathematical Society Publ., 1998.

[24] McCarthy, Michael J., Introduction to Theoretical Kinematics, MIT Press, 1990.

[25] Montana, D., "The Kinematics of Contact and Grasp", in *Int. J. of Rob. Res.*, vol. 7, no. 3, June 1988.

[26] Murray, R., Li, Z., Sastry, S., A Mathematical Introduction to Robotic Manipulation, CRC Press, 1994.

[27] Nahvi, A., Nelson, D., Hollerbach, H., Johnson, D., "Haptic Manipulatin of Virtual Mechanisms from Mechanical CAD Designs," *International Conference on Robotics and Automation*, 1998.

[28] Nelson, D., Johnson, D., Cohen, E., "Haptic Rendering of Surface-to-Surface Sculpted Model Interaction," to appear, Symp. on Haptic Interfaces, ASME International Mechanical Engineering Congress and Exposition, Nashville, TN, Nov. 15-20, 1999.

[29] Nelson, D., Cohen, E., "Interactive Mechanical Design Variation for Haptics and CAD," to appear in *Eurographics*, 1999.

[30] Platt, J., Barr, A., "Constraint Methods for Flexible Models," in *Computer Graphics Proceedings, SIGGRAPH 1988*, pp. 279-288.

[31] Sciavocco, L., Siciliano, B., Modeling and Control of Robot Manipulators, McGraw-Hill, 1996.

[32] Shabana, A., Dynamics of Multibody Systems, Cambridge University Press, 1998.

[33] Snyder, J., Woodbury, A., Fleischer, K., Currin, B., Barr, A., "Interval Methods for Multi-Point Collisions between Time-Dependent Curved Surfaces," in *Computer Graphics Proceedings, SIGGRAPH*, 1993.

[34] Snyder, J., "An Interactive Tool for Placing Curved Surfaces without Interpenetration," in *Computer Graphics Proceedings, SIGGRAPH*, Los Angeles, August 6-11, 1995, pp. 209-216.

[35] Thompson II, T.V., Nelson, D.D., Cohen, E., and Hollerbach, J.M., "Maneuverable NURBS models within a haptic virtual environment," 6th Annual Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems, DSC-Vol. 61, (Dallas, TX), pp. 37-44, Nov. 15-21, 1997.

[36] Wampler, C., "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares method," in *IEEE Transactions on Systems, Man and Cybernetics*, 1(16):93-101, January 1986.

[37] Witkin, A., Fleischer, K., Barr, A., "Energy Constraints on Parameterized Models," in *Computer Graphics Proceedings, SIGGRAPH*, pp. 225-232, 1987.

[38] Zhou, J., Badler, N., "Inverse Kinematics Positioning Using Nonlinear Programming for Highly Articulated Figures," in *Transactions on Graphics*, 13, 4, pp. 313-336, October 1994.

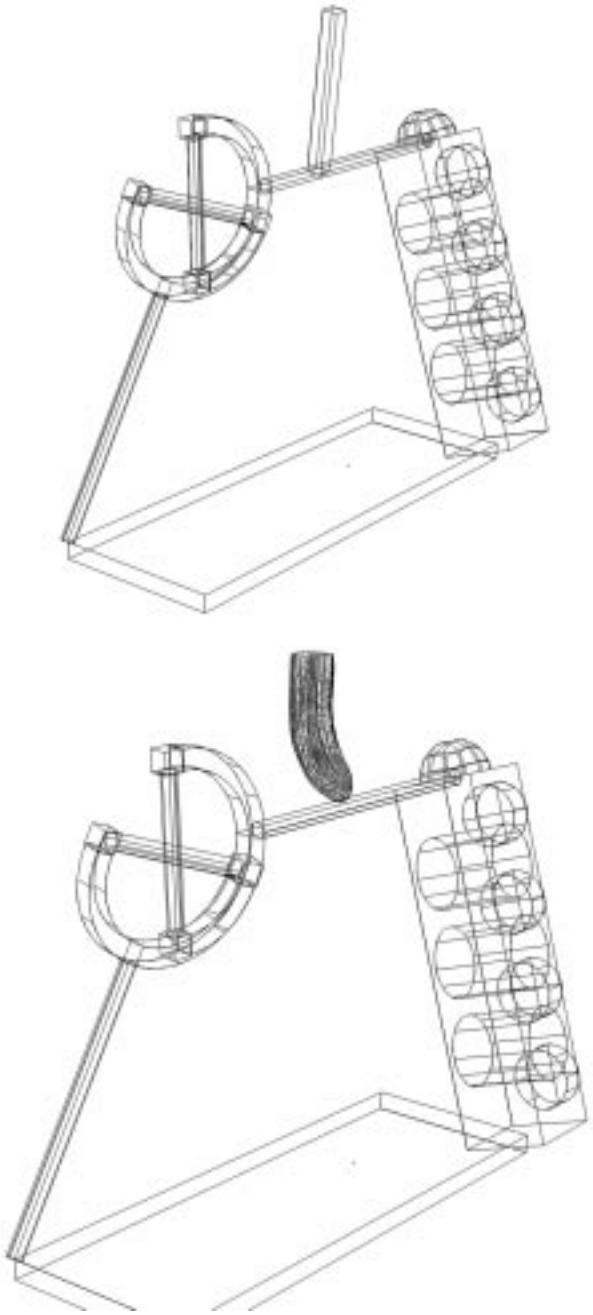

Figure 7: Kinematic manipulation of a spatial four bar mechanism consisting of a universal joint, a spherical joint, and two revolute joints. The objective is to minimize the amount of separation of the middle of the top link with the hand grasp point. The top figure shows a point grasping constraint. The bottom figure has a floating surface-surface contact.