# USER INTERACTION WITH CAD MODELS WITH NONHOLONOMIC PARAMETRIC SURFACE CONSTRAINTS

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

## ABSTRACT

User manipulation of assembly models can provide insight during the early, formulative design stages into kinematic and dynamic characteristics of a mechanism. We present the advantages of kinematic representation of constraint equations in fully Cartesian coordinates, a departure from standard practice for interactive mechanical assembly at interactive rates.

Formulations of a surface rolling contact constraint equation and its Jacobian, defined as a joint between two NURBS surfaces via position, tangency and velocity constraint relations, are derived for use in dynamic simulation and assembly optimization. The constraint equation formulations use quaternions to represent orientation. An appendix develops appropriate differential algebra.

In this work we develop the use of constraints in global frame Cartesian coordinates for describing operator-in-the-loop interactions with mechanical assemblies under a unified framework combining lower-pair joints and more general surface contact interactions.

## 1 Introduction

The kinematics and dynamics of the complicated, highly unconstrained, jointed mechanisms that arise in virtual prototyping are difficult to analyze. Frequently occurring examples include smooth fingered grasp, spatial cam-follower mechanisms, and belt drives.

Constraint equation formulations can be used to support such unusual joints in kinematic optimization problems and dynamics. It is then necessary to decide the form, or *generalized coordinates* with which to express these equations. Generalized coordinates are any set of coordinates used to describe system configuration; further, any quantity may serve as generalized coordinates [Goldstein, 1980], [Shabana, 1994]. The equations applicable to the above cited examples can be expressed in *absolute Cartesian generalized coordinates*, (ACGC) [Shabana, 1994]. *Reduced* (joint-space) *generalized coordinates* may be used as an alternative, but this alternative is not applicable to joints with many degrees of freedom or joints with nonholonomic constraints that cannot be parameterized [Baraff, 1996].

The Newton-Euler equations and other well known dynamics formulations can be expressed in ACGC, and augmented with Lagrange multipliers to account for joint constraint forces [Haug, 1992],[Shabana, 1994]. We have found this approach to have desirable characteristics in software engineering and expressiveness due to modular association of equations with bodies. Also, it is known to have linear cost in the number of constraints [Baraff, 1996]. This technique uses sparse constraint Jacobian matrices, that is, partial derivatives of geometric constraint expressions that enforce the requirement that joints remain connected.

Lagrange multipliers can be used with the parameterization of complicated joints when the joint coordinates are ACGC. For example, the nonholonomic surface rolling contact "joint" as in Fig. 1 can be used to handle velocity-dependent constraints in a framework which manages differential operations on constraints.

The surface contact example shown in Fig. 2 is an interesting case that can come up in kinematic and dynamic analysis of knee or spatial cam mechanisms. The belt drive in Fig. 3 is an example of a planar curve contact. The analyses in Sections 4, 5, and 6 are applicable to both planar curve and spatial surface constraints.

Surface joints will have three position degrees of freedom (Fig. 4). While each body has six degrees of freedom in velocity, three of these are removed in the rolling contact constraints.

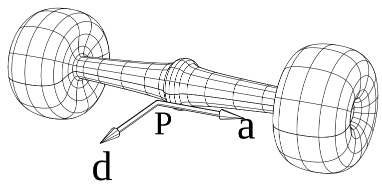


Figure 1. A wheeled vehicle can reach any position and orientation; its position is not constrained. The constraint that can be written is nonholonomic: the velocity of $\mathbf{P}$ along the axle is zero: $\mathbf{a}\cdot\dot{\mathbf{d}} = 0$ [Haug, 1992].

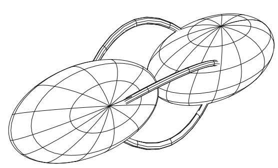

Figure 2. Two links in surface-surface contact, held together.

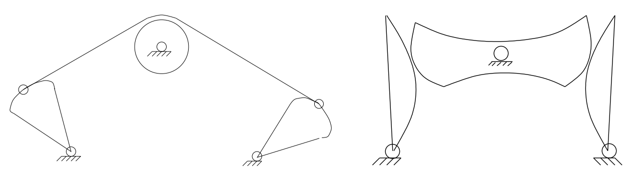

Figure 3. Two links under curve contact constraint. At right is a belt drive transformed into a rolling contact mechanism [Erdman, 1993].

Computational techniques for surface derivatives are presented in Section 5.

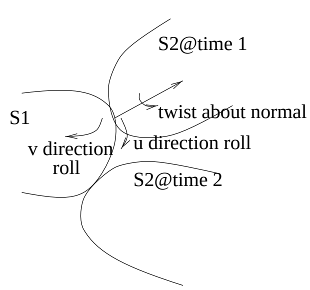


Figure 4. 3 position degrees of freedom are available in a surface contact joint.

## 2 Background on Surface Contact

Many important contact problems are classified as either unilateral or bilateral constraints. A bilateral constraint expresses an equation of equality. The force of constraint in this case can be in either direction of the surface normal. A unilateral constraint expresses inequality and has constraint force in one direction, along the outward surface normal. This work is addressed towards unilateral constraints for dynamics and both unilateral and bilateral constraints for assembly optimization.

Other work has treated surface-surface contacts, or "against surfaces" in terms of a unilateral linear complementarity problem or as a quadratic programming problem [Lotstedt, 1982]. A direct method without optimization is given in [Baraff, 1994]. The nature of the dynamics problem treated here differs in that we are concerned with the geometry of smooth surface joints modeled as bilateral constraints which hold together, rather than unilateral constraints and friction forces of flat or polygonal surface contacts.

Work on rolling contact has studied the problems of grasping, pushing, and steering. Prior work in this area beginning with [Montana, 1988] and [Cai, 1987] has made use of time-dependent relations [Sarkar, 1994], [Yun, 1995], [Canny, 1990], [Cremer, 1996], [Jia, 1998], [Han, 1998]. The evolutionary style of updating the point and angle of contact by adding the parametric contact coordinates to the generalized coordinates is not conveniently used with the Cartesian coordinates in the Lagrange multiplier constraints, formulated in [Shabana, 1994], [Haug, 1992], [Baraff, 1996] and this work. We are concerned with the development of surface contact kinematics in coordinates of the position and orientation of each linkage.

Commercial CAD packages are beginning to incorporate new features in geometric constraints such as edge-to-surface constraints [ProE], [Adams], and modeling of mechanisms with surface constraints is becoming more advanced. However, analysis of such assemblies is very application specific and heavily dependent on particular properties such as planar characteristics of the constraint or motion.

## 3 Linkage Surfaces

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

$$\mathbf{S}(u,v) = \frac{\sum_{i,j} w_{i,j}\mathbf{p}_{i,j}B_{i,k_u}(u)B_{j,k_v}(v)}{\sum_{i,j} w_{i,j}B_{i,k_u}(u)B_{j,k_v}(v)}$$

where the B-spline blending functions $B$, control mesh $\mathbf{p}$, and weights $\mathbf{w}$ are used.

We denote generalized coordinates $\mathbf{q}^i$ to be in Cartesian, world space, for each surface i. Suppose it is necessary to rotate a NURBS linkage by a unit length quaternion 4-tuple containing components of Euler parameters $\mathbf{q}_{4..7}$ and translation by $\mathbf{q}_{1..3}$.

Using the original control mesh, $\mathbf{p}^{orig}$, create the control points of the NURBS surface in the global frame, $\mathbf{p}_{i,j}$. Then $\mathbf{p}_{i,j} = \mathbf{R}(\mathbf{q}_{4..7})\mathbf{p}^{orig}_{i,j} + \mathbf{q}_{1..3}$, where $\mathbf{R}$ is a rotational operator with quaternion arguments. Thus, $\mathbf{p}_{i,j}$ as a function of $\mathbf{q}$, and $\mathbf{S}$ is now a function of $u, v$ and $\mathbf{p}_{0,0}$, $\mathbf{p}_{0,1}$, ... $\mathbf{p}_{n,m}$, and

$$\mathbf{S}(u, v, \mathbf{p}_{0,0}(\mathbf{q}), \mathbf{p}_{0,1}(\mathbf{q}), \ldots) = \frac{\sum_{i,j} w_{i,j}\mathbf{p}_{i,j} B_{i,j_u}(u) B_{j,v_v}(v)}{\sum_{i,j}^{m} w_{i,j} B_{i,k_u}(u) B_{j,k_v}(v)}. \quad (1)$$

## 4 Constraint Equations

This paper presents a modular framework in which simple joints and surface contact joints can be used together for haptics assembly and dynamic interaction. The surface constraints are written in the manner of lower-pair geometric joint constraints as found in [Yang, 1995], [Shabana, 1994], [Haug, 1992]. For example, a spherical joint in Fig. 5 indicates that the endpoints of the two bodies $i$ and $j$ must meet.

$$\mathbf{C}^{sph}(\mathbf{q}) = \mathbf{R}(\mathbf{q}^i_{4..7})\mathbf{k}^i + \mathbf{q}^i_{1..3} - \mathbf{R}(\mathbf{q}^j_{4..7})\mathbf{k}^j - \mathbf{q}^j_{1..3} \quad (2)$$

The constant $\mathbf{k}$ in Eqn. 2 is a local body frame vector indicating the location of the spherical joint (like $\mathbf{p}^{orig}$). The Jacobian of $\mathbf{R}(\mathbf{q}_{i,4..7})$ are used in Section 7 and Section 11.

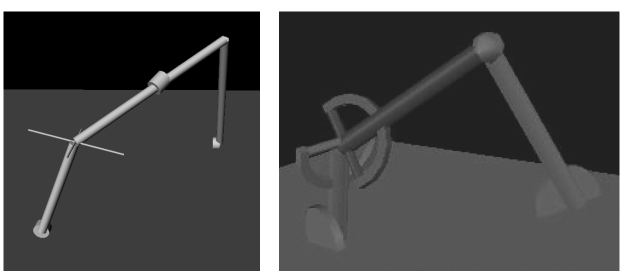

Figure 5. Two differently detailed views of a spatial 1 DOF, 4 bar mechanism with two revolute joints, a spherical joint, and a universal joint. The middle of the middle link is grasped by point contact.

The surface constraint equations express the fact that the surfaces must touch, must be tangent at that point, and (optionally) must roll, not slide around. The position constraint is

$$\mathbf{C}^p = ||\mathbf{S}^1(u^1, v^1) - \mathbf{S}^2(u^2, v^2))||^2 = 0 \quad (3)$$

where $(u^1, v^1)$ and $(u^2, v^2)$ are obtained from a surface-surface optimization algorithm [Kriezis, 1992], [Johnson, 1998]. The two pairs $\{(u^1, v^1), (u^2, v^2)\}$ will be the *parametric contact* coordinates.

Tangency constraints go along with constraint $\mathbf{C}^p$ to enforce that the tangent plane of both surfaces define the same plane.

$$\mathbf{C}^{tan} = \begin{bmatrix} \mathbf{S}^2_u(u^2, v^2) \cdot (\mathbf{S}^1_u(u^1, v^1) \times \mathbf{S}^1_v(u^1, v^1)) \\ \mathbf{S}^2_v(u^2, v^2) \cdot (\mathbf{S}^1_u(u^1, v^1) \times \mathbf{S}^1_v(u^1, v^1)) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (4)$$

Equations $\mathbf{C}^p$ and $\mathbf{C}^{tan}$ are sufficient for a sliding NURBS-NURBS contact condition. In addition, we can write a rolling joint contact constraint. The rolling contact constraint requires that no slippage occur between two surfaces. To prevent dragging or slipping, the amount of movement of the contact point along one surface must be equal to the amount of movement on the other surface. Thus, the relative surface velocity is constrained to reflect this fact. We write the rolling constraint as a time derivative, $\dot{\mathbf{C}}^{roll}$, for consistent notation in Section 8.

We measure the relative change in surface Cartesian contact coordinates as a result of the change in $\mathbf{q}^1$ and $\mathbf{q}^2$ with respect to each body's local frame, or original surface, $\mathbf{S}_L$. For comparison, the relative contact velocity on $\mathbf{S}_L$ is oriented in the world frame and projected onto the contact plane at $\mathbf{S}^1$ containing vectors $\mathbf{S}^1_u$, $\mathbf{S}^1_v$.

$$\begin{aligned} \mathbf{S}^1_u \cdot \mathbf{R}^1(\mathbf{q}^1)\dot{\mathbf{S}}^1_L &= \mathbf{S}^1_u \cdot \mathbf{R}^2(\mathbf{q}^2)\dot{\mathbf{S}}^2_L \\ \mathbf{S}^1_v \cdot \mathbf{R}^1(\mathbf{q}^1)\dot{\mathbf{S}}^1_L &= \mathbf{S}^1_v \cdot \mathbf{R}^2(\mathbf{q}^2)\dot{\mathbf{S}}^2_L \end{aligned} \quad (5)$$

Here the parametric contact coordinates $(u^1, v^1)$ and $(u^2, v^2)$ are also variable over time, and are functions of the position coordinates $\mathbf{q}^1$ and $\mathbf{q}^2$ of each body. The constraint equation that satisfies Eqn. 5, with the substitution $\dot{\mathbf{S}} = \mathbf{R}\dot{\mathbf{S}}_L + \dot{\mathbf{R}}\mathbf{S}_L + \dot{\mathbf{q}}_{1..3}$, is

$$\dot{\mathbf{C}}^{roll} = \begin{bmatrix} \mathbf{S}^1_u \cdot (\dot{\mathbf{S}}^1 - \dot{\mathbf{R}}\mathbf{S}^1_L - \dot{\mathbf{q}}^1{}_{1..3} - \dot{\mathbf{S}}^2 + \dot{\mathbf{R}}\mathbf{S}^2_L + \dot{\mathbf{q}}^2{}_{1..3}) \\ \mathbf{S}^1_v \cdot (\dot{\mathbf{S}}^1 - \dot{\mathbf{R}}\mathbf{S}^1_L - \dot{\mathbf{q}}^1{}_{1..3} - \dot{\mathbf{S}}^2 + \dot{\mathbf{R}}\mathbf{S}^2_L + \dot{\mathbf{q}}^2{}_{1..3}) \end{bmatrix} = \mathbf{0}_{2x1} \quad (6)$$

Each term $\dot{\mathbf{S}}$ reduces to $\frac{\partial \mathbf{S}}{\partial \mathbf{q}}\dot{\mathbf{q}}$.

For the "pure" rolling case, slippage about the axis perpendicular to either equivalent surface contact normal is (optionally) not allowed and removes a third degree of freedom from the velocity characteristics of each body. $(\mathbf{S}^1_u \times \mathbf{S}^1_v) \cdot (\omega^1 - \omega^2) = 0$ may be added as a third equation for $\dot{\mathbf{C}}^{roll}$, where $\omega_1$ and $\omega_2$ are the angular velocities of each body.

## 5 Surface Evaluation Tools: $\frac{\partial \mathbf{S}}{\partial \mathbf{q}}$

To formulate the geometric constraint Jacobians, we must be able to evaluate the surface Jacobian $\frac{\partial \mathbf{S}}{\partial \mathbf{q}}$ use in the formulation of the geometric constraint Jacobians.

Because an interactive rate approximate minimum distance algorithm is available [Kriezis, 1992], [Johnson, 1998] and the analytical evaluation of $\frac{\partial \mathbf{S}}{\partial \mathbf{q}}$ is prohibitive, we use numerical differentiation for finding the Jacobian of the minimal distance equation between two surfaces. For each column $i$ in the Jacobian corresponding to a variable in $\mathbf{q}_i^{1,2} = [\mathbf{q}_i^{1T} \mathbf{q}_i^{2T}]^T$, two samples of the distance equation are evaluated and the gradient is approximated by central finite differences. $\mathbf{S}^1(u^1, v^1, \mathbf{q}^{1,2})$ will be written as $\mathbf{S}^1(\mathbf{q}^{1,2})$ because $(u^1, v^1)$ is obtained from the local distance algorithm and is dependent on $\mathbf{q}^{1,2}$.

$$
\begin{aligned}
\frac{\partial \mathbf{S}^1}{\partial \mathbf{q}_i^{1,2}} &= \frac{\mathbf{S}^1(\mathbf{q}_i^{1,2}+h) - \mathbf{S}^1(\mathbf{q}_i^{1,2}-h)}{2h} \\
\frac{\partial \mathbf{S}^2}{\partial \mathbf{q}_i^{1,2}} &= \frac{\mathbf{S}^2(\mathbf{q}_i^{1,2}+h) - \mathbf{S}^2(\mathbf{q}_i^{1,2}-h)}{2h}
\end{aligned}
\tag{7}
$$

Each column of the Jacobian for each surface requires 2 evaluations of the minimal distance equation [Kriezis, 1992], [Johnson, 1998] for each component of the translations and quaternions in $\mathbf{q}^{1,2}$. This method has error on the order *O($h^2$)*, or $h^2 \frac{\partial^2 \mathbf{S}(\sigma)}{(\partial \mathbf{q})^2}$, for some $\sigma$ in the range $(\mathbf{q}^{1,2} - h, \mathbf{q}^{1,2} + h)$ [Cheney, 1985]. One application of Richardson extrapolation would give an error of *O($h^4$)*, but at the cost of 4 minimal distance evaluations. Some simple numerical analysis shows that a very small value of $h$ relative to the coordinates magnitude may be selected in practice without resulting in loss of precision due to subtraction of very similar numbers. For the purposes of assembly optimization, the *O(h)* formula [Hansen, 1995],

$$
\frac{\partial \mathbf{S}}{\partial \mathbf{q}_i^{1,2}} = \frac{\mathbf{S}(\mathbf{q}_i^{1,2} + h) - \mathbf{S}(\mathbf{q}_i^{1,2})}{h}
$$

has occasionally been inadequate for optimization purposes. The *O($h^2$)* method is used when the *O(h)* technique causes nonsensical results. Accuracy obtained with 4 samples has not been required in examples encountered so far. An additional complication occurs since the quaternion components of $\mathbf{q}$ by $h$ violates the unit length constraint on the quaternion. Renormalization schemes, removal of one of the dependent quaternion components, and the use of Euler angles are possible ways to handle change in rotation coordinates.

The "cross talk" between the surfaces becomes evident here as the partial derivative is with respect to both $\mathbf{q}^1$ and $\mathbf{q}^2$. More intuitively, the motion of one surface will change the closest point on itself and the other surface as well. We note that other polygonal or surface representations that have fast minimal distance evaluation algorithms may use the finite differences algorithm, and that the surface analysis in this paper applies as long as reasonably smooth surface characteristics are a part of the representation.

## 6 Constraint Jacobian

Critical elements needed in performing kinematics optimization and dynamic analysis are the constraint equations and their Jacobians. We derive the partial derivatives of $\mathbf{C}^p$, $\mathbf{C}^{tan}$, and $\dot{\mathbf{C}}^{roll}$, the constraint equations of our approach.

The position constraint Jacobian for $\mathbf{C}^p$ is

$$
\frac{\partial \mathbf{C}^p}{\partial(\mathbf{q}^1, \mathbf{q}^2)} = \frac{\partial(||\mathbf{S}^1(u^1, v^1) - \mathbf{S}^2(u^2, v^2)||^2)}{\partial(\mathbf{q}^1, \mathbf{q}^2)} \tag{8}
$$

and $||\mathbf{S}^1 - \mathbf{S}^2||^2$ is the sum of the squares of the differences of the of the surface evaluations. Recalling that $\frac{\partial \mathbf{S}}{\partial \mathbf{q}}$ is obtainable (Section 5),

$$
\begin{aligned}
&\frac{\partial((\mathbf{S}^1 - \mathbf{S}^2)\cdot(\mathbf{S}^1 - \mathbf{S}^2))}{\partial \mathbf{q}} = \\
&\frac{\partial(\mathbf{S}^1\mathbf{S}^1 - 2\mathbf{S}^1\mathbf{S}^2 + \mathbf{S}^2\mathbf{S}^2)}{\partial \mathbf{q}} = \\
&2\frac{\partial(\mathbf{S}^1)}{\partial \mathbf{q}}\mathbf{S}^1 - 2\left(\frac{\partial(\mathbf{S}^1)}{\partial \mathbf{q}}\mathbf{S}^2 + \frac{\partial(\mathbf{S}^2)}{\partial \mathbf{q}}\mathbf{S}^1\right) + 2\frac{\partial(\mathbf{S}^2)}{\partial \mathbf{q}}\mathbf{S}^2,
\end{aligned}
$$

The algebra for the Jacobian of the tangency constraint is made straightforward by the triple scalar product relation with the determinant of a matrix.

$$
\mathbf{a} \cdot \mathbf{b} \times \mathbf{c} = \begin{vmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{vmatrix} \tag{9}
$$

where we set $\mathbf{a} = \mathbf{S}_u^2(u^2, v^2)$, $\mathbf{b} = \mathbf{S}_u^1(u^1, v^1)$ and $\mathbf{c} = \mathbf{S}_v^1(u^1, v^1)$ from Eqn. 4. Now $\mathbf{C}^{tan}$ becomes, for $\mathbf{d} = \mathbf{S}_v^2(u^2, v^2)$,

$$
\mathbf{C}^{tan} = \begin{bmatrix} a_1(b_2c_3 - b_3c_2) - a_2(b_1c_3 - b_3c_1) + a_3(b_1c_2 - b_2c_1) \\ d_1(b_2c_3 - b_3c_2) - d_2(b_1c_3 - b_3c_1) + d_3(b_1c_2 - b_2c_1) \end{bmatrix} \tag{10}
$$

and

$$
\frac{\partial \mathbf{C}^{tan}}{\partial \mathbf{q}^{1,2}} = \begin{bmatrix} a_{1q}(b_2c_3 - b_3c_2) + a_1(b_{2q}c_3 + c_{3q}b_2 - b_{3q}c_2 - c_{2q}b_3) \\ \ldots - a_{2q}(b_1c_3 - b_3c_1) - a_2(b_{1q}c_3 + c_{3q}b_1 - b_{3q}c_1 - c_{1q}b_3) \\ \ldots + a_{3q}(b_1c_2 - b_2c_1) + a_3(b_{1q}c_2 + c_{2q}b_1 - b_{2q}c_1 - c_{1q}b_2) \\ \\ d_{1q}(b_2c_3 - b_3c_2) + d_1(b_{2q}c_3 + c_{3q}b_2 - b_{3q}c_2 - c_{2q}b_3) \\ \ldots - d_{2q}(b_1c_3 - b_3c_1) - d_2(b_{1q}c_3 + c_{3q}b_1 - b_{3q}c_1 - c_{1q}b_3) \\ \ldots + d_{3q}(b_1c_2 - b_2c_1) + d_3(b_{1q}c_2 + c_{2q}b_1 - b_{2q}c_1 - c_{1q}b_2) \end{bmatrix} \tag{11}
$$

Other joints have an additional rolling constraint condition,

$$\frac{\partial \dot{\mathbf{C}}^{roll}}{\partial(\mathbf{q}^1, \mathbf{q}^2)} = \mathbf{F_q}\mathbf{G} \tag{12}$$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{S}_{1u} \cdot (\mathbf{S}_{1\mathbf{q}}\dot{\mathbf{q}} - \dot{R}_1\mathbf{S}_{1L} - \dot{\mathbf{q}}^1 - \mathbf{S}_{2\mathbf{q}}\dot{\mathbf{q}} + \dot{R}_2\mathbf{S}_{2L} + \dot{\mathbf{q}}^2) \\ \mathbf{S}_{1v} \cdot (\mathbf{S}_{1\mathbf{q}}\dot{\mathbf{q}} - \dot{R}_1\mathbf{S}_{1L} - \dot{\mathbf{q}}^1 - \mathbf{S}_{2\mathbf{q}}\dot{\mathbf{q}} + \dot{R}_2\mathbf{S}_{2L} + \dot{\mathbf{q}}^2) \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \end{bmatrix}, \tag{13}$$

and $\mathbf{F_q}$ is $\frac{\partial \mathbf{F}}{\partial \mathbf{q}}$. Now note $\dot{\mathbf{C}}^{roll}$ is a function of time as well; therefore the terms $\mathbf{G}_t$, $\mathbf{F}_{\mathbf{q}t}$, and $\ddot{\mathbf{C}}^{roll}$ are required in the equations of motion and follow from Eqns. 12 and 13.

## 7 Assembly Optimization for Haptics

Methods have been developed for solving or optimizing geometric constraints for the purposes, among others, of mechanism assembly and tolerance analysis. Artificial intelligence techniques [Kramer, 1992], constraint propagation, degree of freedom analysis, nonlinear programming, algebraic graph reduction [Bourma, 1995], and dynamic forces [Barr, 1988] have all been used as solution methods.

A user's hand grasping an object may be represented as the ground constraint in an assembly. The solution of the resulting system solves the generalized inverse kinematics problem which allows the user to move pieces of the assembly in an interactive way. A review of the formulation of simple constraints in Cartesian coordinates is contained in [Shabana, 1994].

Automatic assembly optimization for finding mechanical configurations satisfying joint constraints $\mathbf{C}(\mathbf{q}) = 0$, i.e. by Newton's optimization method,

$$\mathbf{C_q}\Delta\mathbf{q} = -\mathbf{C}(\mathbf{q}) \tag{14}$$

is useful in assembly analysis to eliminate the need to piece together assemblies by hand. Solving for $\mathbf{q}$ gives a valid mechanism configuration which is required in Section 8.

An iterative technique to approximating a solution is given by

$$\Delta\mathbf{q} = k\mathbf{C}_{\mathbf{q}}^T(-\mathbf{C}(\mathbf{q})) \tag{15}$$

Intuitively, we may think of Eqn. 15 as a kind of spring. When one uses Lagrange multipliers to enforce constraints, multipliers $\lambda$ give rise to a force in configuration space of $\mathbf{C}_{\mathbf{q}}^T\lambda$. The directions of the constraint force are $\mathbf{C_q}$. If we think of springs trying to enforce the constraints, and make the spring force be proportional to error (error is the constraint manifold $\mathbf{C}$), we would have in general

$$spring force = -k\mathbf{C}_{\mathbf{q}}^T\mathbf{C}$$

When forming $\mathbf{C}$ and $\mathbf{C_q}$, the change in rotation of a vector with respect to the change in the coordinates $\mathbf{q}$ is used. [Shabana, 1994] derives such a term $\frac{\partial \mathbf{R}(\mathbf{q}_{4..6})\mathbf{v}}{\partial \mathbf{q}_{4..6}}$. Optimization methods for geometric satisfaction problems use of $\frac{\partial \mathbf{R}(\mathbf{q})\mathbf{v}}{\partial \mathbf{q}}$ but can have difficulties in most configurations with local minima when Euler angles are used. To circumvent local minima problems, we use a quaternion 4-tuple to represent orientation. The steps for evaluating the rotation operator $\frac{\partial(qvq^*)}{\partial q}$ in $\mathbf{C_q}$ are detailed in the appendix on quaternion differentiation.

Because quaternions rotate around a constant vector, getting from one orientation to another in the optimization of the constraint equations of a mechanism by gradient descent is essentially linearly interpolating between position and quaternion orientations. When Euler angles are used, the gradient used in descending towards a more optimal orientation may become stuck in modes. Those modes can cause frequent local minima but they are avoided with the use of quaternions.

The use of geometric constraints with quaternion coordinates therefore reduces to applying forces somewhat like the methods in [Barr, 1988]. We have found experimentally that a choice of k=0.1 is a high but stable value for the examples encountered so far. This constant is a limit on how fast the assembly process converges to satisfaction. Because there is no direct dependence between the size of the mechanism and k, the constant time algorithm in the next section can be applied.

### 7.1 Massively Parallel Optimization

Eqn. 15 is a sparse matrix multiplied by a column vector. The locations of the dense areas in the sparse matrix are known and generated from the constraint relation between bodies. Each element in $\Delta\mathbf{q}$ may be evaluated on a separate processor by sending a row of $\mathbf{C}_{\mathbf{q}}^T$ and a copy of $\mathbf{C}$ to a different processor. Since each processor has at most two or three non-zero areas in $\mathbf{C}_{\mathbf{q}}^T$, only three areas of multiplication need to occur between $\mathbf{C}_{\mathbf{q}}^T$ and $\mathbf{C}$ in practice. The interdependence of data is not a problem; data can be copied or farmed out to each processor and independent results can be collected on a host processor. The algorithm therefore has the property in that it will run in constant time given enough processors.

## 8 Motion Equation

We couple the equations of motion and two time derivatives of the constraint equation $\mathbf{C}(\mathbf{q},t) = \mathbf{0}$ as in

$$\mathbf{C_q}\dot{\mathbf{q}} = -\mathbf{C}_t$$
$$\mathbf{C_q}\ddot{\mathbf{q}} = -\mathbf{C}_{tt} - (\mathbf{C_q}\dot{\mathbf{q}})_{\mathbf{q}}\dot{\mathbf{q}} - 2\mathbf{C}_{\mathbf{q}t}\dot{\mathbf{q}}$$
$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}_{\mathbf{q}}^T\lambda = \mathbf{Q}_e + \mathbf{Q}_v$$

where $\mathbf{Q}_e$ are external applied forces, and $\mathbf{Q}_v$ are velocity dependent terms [Shabana, 1994]. $\mathbf{C}_{\mathbf{q}}^T\lambda$ is the joint constraint force as shown in [Haug, 1992]. The problem can be solved for $\ddot{\mathbf{q}}$ and $\lambda$ in linear time [Baraff, 1996]. This differential-algebraic system may be integrated with the use of stabilization techniques [Baumgarte, 1972], [Negrut, 1997], [Ascher, 1995], [Potra, 1995], [Campbell, 1995].

It is desirable to have the mass matrix $\mathbf{M}$ and vector $\mathbf{Q}_v$ in terms of quaternion acceleration [Omelyan, 1998]. Steps toward the reformulation of the mass matrix and generalized inertial forces that depend on velocity in terms of quaternions by using the angular velocity relation with quaternions may be derived following the manner in [Haug, 1992].

## 9 Results

The constraint optimization techniques have been developed within a prototypical Matlab environment and are being incorporated into Utah's *Alpha_1* modeling environment [Cohen, 1980]. The constraints for lower-pair joints have been implemented in a high performance C++ compiled form. When the operator is a part of the assembly optimization through a grounded finger, usually single iteration has been sufficient for the interactive reassembly. An update of 30kHz is achieved for small examples, wuch as those shown in Figs. 6, 7.

The surface constraint equations have also been developed under Matlab, and are in the process of being incorporated into a compiled form. Convergence of a curve-curve contact example is shown in Fig. 8. Simpler constraint equations, including revolute, spherical, and prismatic joints have been implemented to test the dynamics support framework.

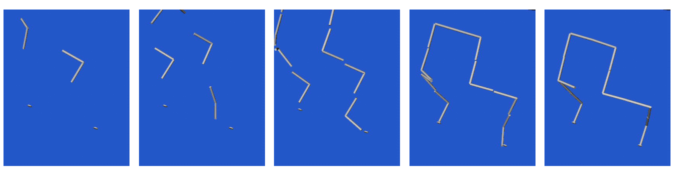

Figure 6. Steps of auto-assembly during optimization for a mechanism with bendy, 90 degree pieces and 2 ground constraints.

The use of gradient descent in the assembly optimization requires more than 15 iterations when links are very far away as in the beginning of a demonstration. However, only 1 iteration per servo cycle is required in the usual case due to the fact that so many updates per second are obtained with the linear time algorithm and that the user's hand cannot move very far in $10^{-5}$ seconds. This is a performance advantage for haptics methods over quadratically converging, cubic cost algorithms [SDFast, 1990], [Garcia, 1994], [Shabana, 1994], [Nahvi, 1998], [Lenarcic, 1998].

The avoidance of local minima for largely unassembled parts is also an advantage as discussed in Section 7. "Nice-looking" solutions are obtained in an attempt to meet the assembly constraints because the optimization "forces" propagate throughout the mechanism.

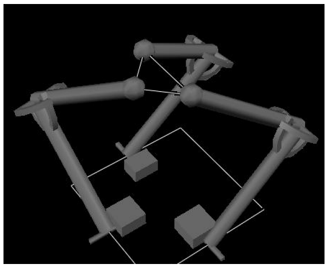

Figure 7. Assembly solution for a Stewart platform.

The assembly optimization procedure applies to other design variables besides link positions. Flexible body coordinates, link length, and other parameters have also been optimized in this framework in a manner similar to [Ashrafiuon, 1990], [Hansen, 1995], [Zou, 1997].

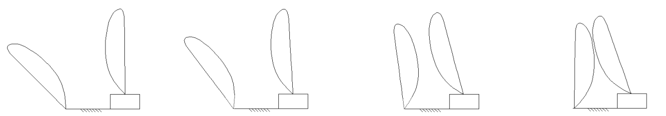

Figure 8. Assembly sequence for curve-curve mechanism [Erdman, 1993]. Both links are also constrained by two revolute joints attached to ground, making it a 1 DOF planar mechanism.

## 10 Conclusion

The approach advocated in this paper for constraint satisfaction, inverse kinematics, and similar assembly problems is the use of constraint equations and the Jacobian of constraint equations. The advantage of this approach is a compact implementation, re-use of constraints for dynamics, computation expense linear in the number of constraints, and massive parallelism for constant asymptotic running time.

The formulation of surface rolling constraints in the framework of constraint Jacobians for use in assembly optimization and Lagrange multiplier dynamics has been derived. Initial results demonstrated both for this formulation have been shown to have a tractable parameterization and small implementation. Additional research is needed to test the scalable characteristics of the approach on larger mechanisms.

## 11 Appendix: Quaternion Differential Algebra

The computational cost of memory caching and multiplications in the operations performed by rotation matrices, Euler angles, Rodriguez parameters, and quaternions are all similar [Funda, 1990]. But the unique representation of orientation is best accomplished with quaternions.

The quaternion rotation operates on a vector by:

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

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

The rotation constraint of a unit quaternion is that $||q_o + q_x^2 + q_y^2 + q_z^2|| = 1$. We could eliminate $q_0$ as dependent with $q_0 = (1 - q_x^2 - q_y^2 - q_z^2)^{\frac{1}{2}}$ at this point, but for both dynamics and assembly considerations, the unit length constraint is more easily introduced in the geometric constraint equation and Jacobian. This allows us to ignore the special case that occurs when the scalar quaternion part is 0 and to obtain a simple differentiation operation with all four elements that avoids the square root introduced above. This important operation used in simple joints and more complicated surface constraints is now given by

$$\frac{\partial(qvq^*)}{\partial q} = \left[\begin{matrix} q_y v_z - q_z v_y & q_x v_x + q_y v_y + q_z v_z & \dots \\ q_0 v_y + q_z v_x - q_x v_z & -q_x v_y - q_0 v_z + q_y v_x & \dots \\ q_0 v_z + q_x v_y - q_y v_x & q_0 v_y + q_z v_x - q_x v_z & \dots \end{matrix}\right.$$

$$\left.\begin{matrix} q_0 v_z + q_x v_y - q_y v_x & -q_z v_x - q_0 v_y + q_x v_z \\ q_x v_x + q_y v_y + q_z v_z & -q_z v_y + q_0 v_x + q_y v_z \\ -q_y v_z - q_0 v_x + q_z v_y & q_x v_x + q_y v_y + q_z v_z \end{matrix}\right] \qquad (16)$$

## Acknowledgments

The authors thank David Johnson, Thomas Thompson, and Ali Nahvi for editing help. Thanks also go to the students and staff of the *Alpha_1* project, within which this work was developed. Support for this research was provided by NSF Grant MIP-9420352, by DARPA grant F33615-96-C-5621, and by the NSF and DARPA Science and Technology Center for Computer Graphics and Scientific Visualization (ASC-89-20219).

## REFERENCES

[Adams] Adams Reference Manual, Adams/Solver, Mechanical Dynamics, Inc., Ann Arbor MI, 1991.

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

[Ascher, 1995] Ascher, U., Chin, H., Petzold, L., Reich, S., "Stabilization of Constrained Mechanical Systems with DAEs and Invariant Manifolds," *Mech. Struct. & Mach.*, vol. 23, no. 2, pp. 135-157, 1995.

[Ashrafiuon, 1990] Ashrafiuon, H., Mani, N., "Analysis and Optimal Design of Spatial Mechanical Systems," in Jour. of Mech. Des., vol. 112, no. 2., June 1990.

[Baraff, 1994] Baraff, D., "Fast Contact Force Computation for Nonpenetraing Rigid Bodies," in *Computer Graphics Proceedings, SIGGRAPH*, Orlando, July 24-29, 1994.

[Baraff, 1996] Baraff, D., "Linear Time Dynamics Using Lagrange Multipliers," in *Computer Graphics Proceedings, SIGGRAPH*, New Orleans, August 4-9, 1996.

[Barr, 1992] Barr, A., Currin, G., Gabriel, S., Hughes, J., "Smooth Interpolation of Orientations with Angular Velocity Constraints using Quaternions," in *Computer Graphics Proceedings, SIGGRAPH*, Chicago, July 26-31, pp. 313-320, 1992.

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

[Baumgarte, 1972] Baumgarte, J., "Stabilization of constraints and integrals of motion in dynamical systems," in *Comp. Meth. in Appl. Mech. and Eng.*,, vol 1, pp 1-16, 1972.

[Bourma, 1995] Bouma, W., Fudos, I., Hoffmann, C.M., "A Geometric Constraint Solver," in *CAD*, 27, 1995, pp. 487-501.

[Campbell, 1995] Campbell, S., "High-Index Differential Algebraic Equations," *Mech. Struct. & Mach.*, vol. 23, no. 2, pp. 199-222, 1995.

[Cai, 1987] 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.

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

[Cohen, 1980] 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.

[Denavit, 1955] Denavit, J., Hartenberg, R., "A Kinematic Notation for Lower Pair Mechanisms based on Matrices," *ASME Jour. of Appl. Mech.*, vol. 22E, 1955.

[McCarthy, 1991] Dooley, J. R., McCarthy, J. M., "Spatial rigid body dynamics using dual quaternion components," in *IEEE Int. Conf. on Robotics and Automation*, April, 1991, vol. 1, 1991.

[Erdman, 1993] Erdman, A., Editor, *Modern Kinematics: Developments in the Last Forty Years*, John Wiley & Sons, 1993.

[Funda, 1990] Funda, J., Taylor, R.H., and Paul, R.P., "On homogeneous transforms, quaternions, and computational efficiency," in *IEEE Trans. Autom. and Rob.*, vol 6, pp. 382-388, 1990.

[Garcia, 1994] Garcia de Jalon, J., et. al., "Kinematic and Dynamic Simulation of Rigid and Flexible Systems with Fully Cartesian Coordinates," in Computer-Aided Analysis of Rigid and Flexible Mechanical Systems, Pereira, Ambrosio, editors, Kluwer Academic Publishers, pp. 290-292, 1994.

[Goldstein, 1980] Goldstein, H., Classical Mechanics, 2nd ed., Addison-Wesley, 1980.

[Han, 1998] Han, L., Trinkle, J., "The Instantaneous Kinematics of Manipulation," in *Int. Conf. on Robotics and Automation*, pp. 1944-1949., 1998.

[Hansen, 1995] Hansen, J., "Synthesis of Spatial Mechanisms Using Optimization and Continuation Methods," Periera, M., Ambrosio, J., eds, Computational Dynamics in Multibody Systems, Kluwer Academic Publishers, 1995.

[Haug, 1992] Haug, E., Interm. Dynamics, Prentice Hall, 1992.

[Johnson, 1998] Johnson, D., Dissertation, Univ. of Utah, 1998.

[Jia, 1998] Jia, Y., Erdmann, M., "Observing Pose and Motion through Contact," in *Int. Conf. Robotics and Automation*, pp. 723-729., 1998.

[Jones, 1997] Jones, R., "Constraint-Based Interactive Assembly Planning," *Int. Conf. on Robotics and Automation*, pp. 913-918. 1997.

[Kramer, 1992] Kramer, G.A., "A Geometric Constraint Engine," in *Artificial Intelligence*, 58, pp. 327-360, 1992.

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

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

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

[Lotstedt, 1982] Lotstedt, P., "Mechanical systems of rigid bodies subject to unilateral constraints," in *SIAM Jour. of Appl. Math.*, vol. 42, pp. 281-296, April 1992.

[Martin, 1969] Martin, G., Kinematics and Dynamics of Machines, McGraw-Hill, 1969.

[Maekawa, 1997] Maekawa, H., Tanie, K., Komoriya, K., "Kinematics, Statics, and Stiffness Effect of 3D Grasp by Multifingered Hand with Rolling Contact at the Fingertip," in *Int. Conf. on Robotics and Automation*, pp. 78-83.

[Metaxas, 1992] Metaxas, D., Terzopoulos, D., "Dynamic Deformation of Solid Primitives with Constraints," in *Computer Graphics Proceedings, SIGGRAPH*, Chicago, July 26-31, 1992.

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

[Nahvi, 1998] Nahvi, A., Nelson, D., Hollerbach, H., Johnson, D., "Haptic Manipulation of Virtual Mechanisms from Mechanical CAD Designs," *Int. Conf. on Robotics and Automation*, 1998.

[Negrut, 1997] Haug, E., Negrut, D., Iancu, M., "A State-Space-Based Implicit Integration Algorithm for Differential-Algebraic Equations of Multibody Dynamics," *Mech. Struct. & Mach.*, vol. 25., pp. 311-334, 1997.

[Omelyan, 1998] Omelyan, I., "On the numerical integration of motion for rigid polyatomics: The modified quaternion approach," *Computers in Physics*, Feb. 1998.

[Papalambros, 1995] Papalambros, P., "Optimal Design of Mechanical Engineering Systems," in *Jour. of Mech. Des.* vol. 117, no. 2, June 1995.

[Petzold, 1994] Petzold, L., "Computational Challenges in Mechanical Systems Formulation," in Computer-Aided Analysis of Rigid and Flexible Mechanical Systems, Pereira, Ambrosio, editors, Kluwer Academic Publishers, 1994.

[ProE] Pro/Engineer User's Manual, version 18.

[Potra, 1995] Potra, F., "Runge-Kutta Integrators for Multibody Dynamics," *Mech. Struct. & Mach.*, vol. 23, no. 2, pp. 181-197, 1995.

[Sarkar, 1994] Sarkar, N., Yun, X., Kumar, V., "Control of Mechanical Systems with Rolling Constraints: Application to Dynamic Control of Mobile Robots," in *Int. Jour. of Rob. Res.*, vol 13, no. 1, Feb. 1994, pp. 55-69.

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

[SDFast, 1990] SDFast user's manual, Symbolic Dynamics, 1990.

[Shabana, 1994] Shabana, A., Computational Dynamics, John Wiley & Sons, New York, NY, 1994.

[Shiller, 1993] Shiller, Z., Sundar, S., "Design of Multi-Degree-of-Freedom Mechanisms for Optimal Dynamic Performance," in *Jour. of Mech. Des.*, vol. 115, no. 2, 1993.

[Shoemake, 1985] Shoemake, D., "Animating rotation with quaternion curves," in *Computer Graphics Proceedings, SIGGRAPH*, pp. 245-254, July, 1985.

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

[Yang, 1995] Yang, K., Lee, S., Bae, D., Suh, M., "Use of Joint Geometric Conditions in Formulating Cartesian Constraint Equations," *Mech. Struct. & Mach.*, vol. 23, no. 3, pp. 395-417, 1995.

[Yun, 1995] Yun, X., "State Space Representation of Holonomic and Nonholonomic Constraints Resulting from Rolling Contacts," in *IEEE Int. Conf. on Robotics and Automation*, 1995, pp. 2690-2694.

[Zou, 1997] Zou, H., Abdel-Malek. K., Wang, J., "A Variational Approach for the Design of Spatial Four-Bar Mechanism," *Mech. Struct. & Mach.*, vol. 25., pp. 41-59, 1997.