On the Computation of Rigid Body Motion

CALIN BELTA and VIJAY KUMAR*
{*calin* | *kumar*}*@grasp.cis.upenn.edu*
General Robotics, Automation, Sensing and Perception Laboratory
University of Pennsylvania, Philadelphia, PA, USA

Abstract

We present two methods for generating trajectories and motion interpolants that minimize energy functionals on $SE(3)$, the group of rigid body displacements. We treat $SE(3)$ as a Lie Group and endow it with a left-invariant Riemannian metric derived from energy considerations. The first method is based on a relaxation formulation where the Euler-Lagrange equations for the minimization problem are solved. The second method offers a closed form, but approximate solution to the problem. Both methods preserve invariance. We compare the two methods in terms of computational efficiency and accuracy.

1 Introduction

We consider methods for finding a smooth interpolating motion between two given positions and orientations of a rigid body. This problem finds applications in robotics and computer graphics. The problem is well understood in Euclidean spaces [1, 2]. However, the group of all rigid body displacements, $SE(3)$, is a non-Euclidean space. It is desirable that the computational scheme be independent of the description of the space and be invariant with respect to the choice of the coordinate systems used to describe the motion. Secondly, the smoothness properties and the optimality of the trajectories need to be considered. It is necessarily to endow $SE(3)$ with a metric, and derive trajectories that minimize energy measures derived from this metric.

Shoemake [3] proposed a scheme for interpolating rotations with Bezier curves based on the spherical analog of the de Casteljau algorithm. The focus in this article is on the generalization of the notion of interpolation from the Euclidean space to a curved space, and invariance properties or metrics are not specifically considered. The basic idea was extended by Ge and Ravani [4] and Park and Ravani [5] to spatial motions. Park and Ravani [5] use a left invariant metric on $SE(3)$ for this generalization. However, there is no optimality result in this paper.

Another class of methods is based on the representation of Bezier curves with Bernstein polynomials. Ge and Ravani [6] used the dual unit quaternion representation of $SE(3)$ and subsequently applied Euclidean methods to interpolate in this space. Jütler [7] formulated a more general version of the polynomial interpolation by using dual (instead of dual unit) quaternions to represent $SE(3)$. In such a representation, an element of $SE(3)$ corresponds to a whole equivalence class of dual quaternions. Park and Kang [8] derived a rational

*3401 Walnut St., Philadelphia, PA 19104-6228, Tel:(215) 898-5814

interpolating scheme for the group of rotations $SO(3)$ by representing the group with Cayley parameters and using Euclidean methods in this parameter space. The advantage of these methods is that they produce rational curves. However, once again, none of these papers emphasize the optimality of the resulting motion.

It is also worth noting that all these works (with the exception of [5]) use a particular coordinate representation of the group. In contrast, Noakes *et al.* [9] derived the necessary conditions for cubic splines on general manifolds without using a coordinate chart. These results are extended in [10] to the dynamic interpolation problem. Necessary conditions for higher-order splines are derived in Camarinha *et al.* [11]. A coordinate free formulation of the variational approach was used to generate shortest paths and minimum acceleration and jerk trajectories on $SO(3)$ and $SE(3)$ in [12]. However, analytical solutions are available only in the simplest of cases, and the procedure for solving optimal motions, in general, is computationally intensive. If optimality is sacrificed, it is possible to generate bi-invariant trajectories for interpolation and approximation using the exponential map on the Lie Algebra [13]. While the solutions are of closed-form, the resulting trajectories have no optimality properties.

This paper is built on the results from [12, 13, 14] to generate smooth curves for general Riemannian metrics. We present two methods for generating trajectories and motion interpolants that minimize energy functionals on $SE(3)$, the group of rigid body displacements. The first method is a straightforward extension of the results derived in [9, 12]. We use a relaxation formulation to solve the Euler-Lagrange equations for the minimization problem. Our second method offers a closed form, but approximate solution to the problem. Both methods preserve invariance. We present examples to compare the two methods in terms of computational efficiency and accuracy.

2 Kinematics and Differential Geometry

2.1 The Lie Groups $SO(3)$ and $SE(3)$

Let $GL(3)$ denote the general linear group of dimension 3. As a manifold, $GL(3)$ can be regarded as an open subset of $\mathbb{R}^9$. Moreover, matrix multiplication and inversion are both smooth operations, which make $GL(3)$ a Lie group. The special orthogonal group is a subgroup of the general linear group, defined as

$$SO(3) = \left\{ R \in GL(3) \,|\, RR^T = I, \mathrm{detR} = 1 \right\}$$

$SO(3)$ is referred to as the rotation group on $\mathbb{R}^3$. $GA(3) = GL(3) \times \mathbb{R}^3$ is the affine group. $SE(3) = SO(3) \times \mathbb{R}^3$ is the special Euclidean group, and is the set of all rigid displacements in $\mathbb{R}^3$.

Consider a rigid body moving in free space. Assume any inertial reference frame $\{F\}$ fixed in space and a frame $\{M\}$ fixed to the body at point O' as shown in Figure 1. At each instance, the configuration (position and orientation) of the rigid body can be described by a homogeneous transformation matrix, A, corresponding to the displacement from frame $\{F\}$ to frame $\{M\}$. $SE(3)$ is the set of all rigid body transformations in three-dimensions:

$$SE(3) = \left\{ A \mid A = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}, R \in \mathbb{R}^{3\times 3}, d \in \mathbb{R}^3, R^T R = I, \det(R) = 1 \right\}.$$

$SE(3)$ is a closed subset of $GA(3)$, and, therefore, a Lie group.

On any Lie group the tangent space at the group identity has the structure of a Lie algebra. The Lie algebras of $SO(3)$ and $SE(3)$, denoted by $so(3)$ and $se(3)$ respectively, are given by:

$$so(3) = \left\{ \hat{\omega} \in \mathbb{R}^{3\times 3}, \; \hat{\omega}^T = -\hat{\omega} \right\}, \; se(3) = \left\{ \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix} \mid \hat{\omega} \in \mathbb{R}^{3\times 3}, v \in \mathbb{R}^3, \, \hat{\omega}^T = -\hat{\omega} \right\}.$$

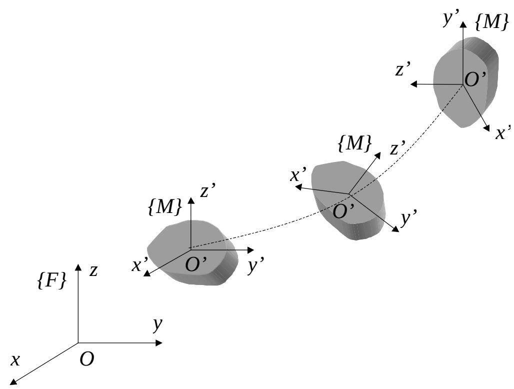

Figure 1: The inertial (fixed) frame and the moving frame attached to the rigid body

A 3×3 skew-symmetric matrix $\hat{\omega}$ can be uniquely identified with a vector $\omega \in \mathbb{R}^3$ so that for an arbitrary vector $x \in \mathbb{R}^3$, $\hat{\omega}x = \omega \times x$, where $\times$ is the vector cross product operation in $\mathbb{R}^3$. Each element $S \in se(3)$ can be thus identified with a vector pair $\{\omega, v\}$. Given a curve

$$A(t) : [-a, a] \rightarrow SE(3),\ A(t) = \left[\begin{array}{cc} R(t) & d(t) \\ 0 & 1 \end{array} \right]$$

an element $S(t)$ of the Lie algebra $se(3)$ can be associated to the tangent vector $\dot{A}(t)$ at an arbitrary point t by:

$$S(t) = A^{-1}(t)\dot{A}(t) = \left[\begin{array}{cc} \hat{\omega}(t) & R^T\dot{d} \\ 0 & 0 \end{array} \right]. \tag{1}$$

where $\hat{\omega}(t) = R^T\dot{R}$ is the corresponding element from $so(3)$.

A curve on $SE(3)$ physically represents a motion of the rigid body. If $\{\omega(t), v(t)\}$ is the vector pair corresponding to $S(t)$, then ω physically corresponds to the angular velocity of the rigid body while v is the linear velocity of the origin O' of the frame $\{M\}$, both expressed in the frame $\{M\}$. In kinematics, elements of this form are called twists and $se(3)$ thus corresponds to the space of twists. The twist $S(t)$ computed from Equation (1) does not depend on the choice of the inertial frame $\{F\}$. For this reason, $S(t)$ is called the left invariant representation of the tangent vector $\dot{A}$. Alternatively, the tangent vector $\dot{A}$ can be identified with a right invariant twist (invariant with respect to the choice of the body-fixed frame $\{M\}$).

Since $so(3)$ is a vector space, any element can be expressed as a 3×1 vector of components corresponding to a chosen basis. The standard basis for $so(3)$ is:

$$L_1^o = \hat{e}_1,\ \ L_2^o = \hat{e}_2,\ \ L_3^o = \hat{e}_3 \tag{2}$$

where

$$e_1 = \left[\begin{array}{ccc} 1 & 0 & 0 \end{array} \right]^T,\ e_2 = \left[\begin{array}{ccc} 0 & 1 & 0 \end{array} \right]^T,\ e_3 = \left[\begin{array}{ccc} 0 & 0 & 1 \end{array} \right]^T$$

is the canonical basis of $\mathbb{R}^3$. L_1^o, L_2^o and L_3^o represent instantaneous rotations about the Cartesian axes x, y and z, respectively. The components of a $\hat{\omega} \in se(3)$ in this basis are given precisely by the angular velocity vector ω.

The standard basis for $se(3)$ is:

$$L_1 = \begin{bmatrix} L_1^o & 0 \\ 0 & 0 \end{bmatrix} \quad L_2 = \begin{bmatrix} L_2^o & 0 \\ 0 & 0 \end{bmatrix} \quad L_3 = \begin{bmatrix} L_3^o & 0 \\ 0 & 0 \end{bmatrix}$$
$$L_4 = \begin{bmatrix} 0 & e_1 \\ 0 & 0 \end{bmatrix} \quad L_5 = \begin{bmatrix} 0 & e_2 \\ 0 & 0 \end{bmatrix} \quad L_6 = \begin{bmatrix} 0 & e_3 \\ 0 & 0 \end{bmatrix} \tag{3}$$

The twists L_4, L_5 and L_6 represent instantaneous translations along the Cartesian axes x, y and z, respectively. The components of a twist $S \in se(3)$ in this basis are given precisely by the velocity vector pair, $\{\omega, v\}$.

2.2 Left Invariant Vector Fields

A *differentiable vector field* is a smooth assignment of a tangent vector to each element of the manifold. At each point, a vector field defines a unique *integral curve* to which it is tangent [15]. Formally, a vector field X is a (derivation) operator which, given a differentiable function, returns its derivative (another function) along the integral curves of X.

An example of a differentiable vector field, X, on $SE(3)$ is obtained by left translation of an element $S \in se(3)$. The value of the vector field X at an arbitrary point $A \in SE(3)$ is given by:

$$X(A) = \bar{S}(A) = AS. \tag{4}$$

A vector field generated by Equation (4) is called a left invariant vector field and we use the notation $\bar{S}$ to indicate that the vector field was obtained by left translating the Lie algebra element S. Right invariant vector fields can be defined analogously. By construction, the set of left or right invariant vector fields is isomorphic to the Lie algebra $se(3)$.

Since the vectors $L_1, L_2, \ldots, L_6$ are a basis for the Lie algebra $se(3)$, the vectors $\bar{L}_1(A), \ldots, \bar{L}_6(A)$ form a basis of the tangent space at any point $A \in SE(3)$. Therefore, any vector field X can be expressed as $X = \sum_{i=1}^{6} X^i \bar{L}_i$, where the coefficients X^i vary over the manifold. If the coefficients are constants, then X is left invariant. By defining $\omega = [X^1, X^2, X^3]^T$ and $v = [X^4, X^5, X^6]^T$, we can associate a vector pair of functions $\{\omega, v\}$ to an arbitrary vector field X. If a curve $A(t)$ describes a motion of the rigid body and $V = \frac{dA}{dt}$ is the vector field tangent to $A(t)$, the vector pair $\{\omega, v\}$ associated with V corresponds to the instantaneous twist (screw axis) for the motion. In general, the twist $\{\omega, v\}$ changes with time.

2.3 Exponential Map and Local Parameterization of $SE(3)$

For every $S \in se(3)$, let $A_S : \mathbb{R} \to SE(3)$ denote the integral curve of the left invariant vector field $\bar{S}$ passing through I at $t = 0$. That is, $A_S(0) = I$ and

$$\frac{d}{dt} A_S(t) = \bar{S}(A_S(t)) = A_S(t) S$$

It follows that [15] $A_S(s+t) = A_S(t)A_S(s)$ which means that $A_S(t)$ is a one-parameter subgroup of SE(3). The function $\exp : se(3) \to SE(3)$ defined by $\exp(S) = A_S(1)$ is called *the exponential map* of the Lie algebra $se(3)$ into $SE(3)$. It is easy to show [16] that the exponential map takes the line $tS \in se(3),\ t \in \mathbb{R}$ into a one-parameter subgroup of $SE(3)$, i.e.

$$\exp(tS) = A_S(t). \tag{5}$$

Using Equation (1) we can show that the exponential map agrees with the usual exponentiation of the matrices in $\mathbb{R}^{4\times 4}$:

$$\exp(tS) = \sum_{k=0}^{\infty} \frac{t^k \mathrm{S}^k}{k!}, \tag{6}$$

where S denotes the matrix representation of the twist S. The sum of this series can be computed explicitly and the resulting expression, when restricted to $SO(3)$, is known as Rodrigues' formula. The formula for the sum in $SE(3)$ is derived in [16].

In this paper, we choose a parameterization of $SE(3)$ induced by the product structure $SO(3) \times \mathbb{R}^3$. In other words, we define a set of coordinates σ_1, σ_2, σ_3, d_1, d_2, d_3 for an arbitrary element $A \in SE(3)$:

$$A = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}$$

so that d_1, d_2, d_3 are the coordinates of d in $\mathbb{R}^3$. Exponential coordinates are chosen as local parameterization of $SO(3)$. For $R \in SO(3)$ sufficiently close to the identity (i.e excluding the points $Tr(R) = -1$ ($Tr(A) = 0$), or, equivalently, rotations through angles of π), we define the exponential coordinates:

$$R = \exp(\hat{\sigma}) = e^{\hat{\sigma}},\ \sigma \in \mathbb{R}^3$$

where $\hat{\sigma}$ is the skew-symmetric matrix corresponding to $\sigma = [\sigma_1,\ \sigma_2,\ \sigma_3]^T$.

The time derivative of σ can be expressed in terms of the body velocity as [17]:

$$\dot{\sigma} = \left(I + \frac{1}{2}\hat{\sigma} + (1 - \alpha(\|\sigma\|)) \frac{\hat{\sigma}^2}{\|\sigma\|^2} \right) \omega \tag{7}$$

where $\alpha(y) = (y/2)cot(y/2)$.

2.4 Riemannian Metrics on $SE(3)$

If a smoothly varying, positive definite, bilinear, symmetric form $< .,. >$ is defined on the tangent space at each point on the manifold, such a form is called a Riemannian metric and the manifold is Riemannian [15]. If the form is non-degenerate but indefinite, the metric is called semi-Riemannian. On a n dimensional manifold, the metric is locally characterized by a $n \times n$ matrix of $\mathcal{C}^\infty$ functions $< X_i, X_j >$ where X_i are basis vector fields.

On $SE(3)$ (on any Lie group), an inner product on the Lie algebra can be extended to a Riemannian metric over the manifold using left (or right) translation.

A metric that is attractive for trajectory planning can be obtained by considering the dynamic properties of the rigid body. The kinetic energy of a rigid body is a scalar that does not depend on the choice of the inertial reference frame. It thus defines a left invariant metric. For this metric, $\frac{1}{2} < V, V >$ corresponds to the kinetic energy of the rigid body moving with a velocity V. If the body-fixed reference frame is attached at the centroid and aligned with the principal axes, then the matrix of this metric is given by:

$$\begin{bmatrix} G & 0 \\ 0 & mI \end{bmatrix}, \tag{8}$$

where m is the mass of the rigid body and G is the inertia matrix:

$$G = \mathrm{diag}\{G_{xx},\ G_{yy},\ G_{zz}\}$$

with G_{xx}, G_{yy}, and G_{zz} denoting the moments of inertia about the x, y, and z axes, respectively. If $\{\omega, v\}$ is the vector pair associated with the vector V, then its norm assumes the familiar expression:

$$< V, V > = \omega^T G \omega + m v^T v. \tag{9}$$

3 Optimal Curves on $SE(3)$

In this section we summarize some results derived in [12].

3.1 Minimum Distance (Energy) Curves - Geodesics

The geodesics can be defined as minimum length curves. The length of a curve $A(t)$ between the points $A(a)$ and $A(b)$ is defined to be:

$$L(A) = \int_a^b < V, V >^{\frac{1}{2}} dt \tag{10}$$

where $V = \frac{dA(t)}{dt}$. It can be shown [15], that if there exists a curve that minimizes the functional L, this curve also minimizes the so called *energy functional*:

$$E(A) = \int_a^b < V, V > dt \tag{11}$$

In [12] it has been proved that a geodesic $A(t)$ on $SE(3)$ equipped with metric (8) is described by

$$\frac{d\omega}{dt} = -G^{-1}(\omega \times (G\omega)) \tag{12}$$

$$\ddot{d} = 0. \tag{13}$$

If $G = \alpha I$, an analytical expression for the geodesic passing through

$$A(0) = \begin{bmatrix} R(0) & d(0) \\ 0 & 1 \end{bmatrix}, \quad A(1) = \begin{bmatrix} R(1) & d(1) \\ 0 & 1 \end{bmatrix} \tag{14}$$

at $t = 0$ and $t = 1$ respectively, is given by [12]:

$$A(t) = \begin{bmatrix} R(t) & d(t) \\ 0 & 1 \end{bmatrix} \in SE(3) \tag{15}$$

where

$$R(t) = R(0)\exp(\hat{\omega}_0 t) \tag{16}$$

$$d(t) = (d(1) - d(0))\, t + d(0) \tag{17}$$

$$\hat{\omega}_0 = \log(R(0)^T R(1)) \tag{18}$$

In the case when $G \neq \alpha I$, there is no closed form expression for the corresponding geodesic and numerical methods or the projection method [14] should be employed.

3.2 Minimum Acceleration Curves

The necessary conditions for the curves that minimize the square of the L^2 norm of the acceleration are derived by considering the first variation of the acceleration functional

$$L_a = \int_a^b < \nabla_V V, \nabla_V V > dt, \tag{19}$$

where $V(t) = \frac{dA(t)}{dt}$, $A(t)$ is a curve on the manifold, and ∇ is the unique symmetric connection compatible with the given metric. The initial and final point as well as the initial and final velocity for the motion are prescribed.

If $SE(3)$ is equipped with metric (8) and $G = \alpha I$, the differential equations to be satisfied by a minimum acceleration curve (15) are [12]:

$$\omega^{(3)} + \omega \times \ddot{\omega} = 0 \quad (20)$$
$$d^{(4)} = 0, \quad (21)$$

As observed in [9], equation (20) can be integrated to obtain

$$\omega^{(2)} + \omega \times \dot{\omega} = \text{constant}$$

However, this equation cannot be further integrated analytically for arbitrary boundary conditions. In [12] it is shown that for special choice of the initial and final velocities, minimum acceleration curves are reparameterized geodesics. If $G \neq \alpha I$ in metric (8), the differential equations to be satisfied by the minimum acceleration curves are difficult to derive and not suited for numerical integration.

4 Motion Generation

As shown in the previous section, closed form solutions for optimal curves on $SE(3)$ with metric (8) are known only for geodesics with $G = \alpha I$. In all the remaining cases, numerical methods (shooting, relaxation) should be employed. We implemented two methods to generate smooth rigid body motion: the relaxation method (Section 4.1) and the projection method (Section 4.2). For the relaxation method, the local chart defined in Section 2.3 is used to parameterize $SE(3)$. The projection method uses the entries in the matrix as coordinates.

4.1 Relaxation Method

The relaxation method [18] is used to generate geodesics when $G \neq \alpha I$ and minimum acceleration curves when $G = \alpha I$. The translational parts (13) and (21) are easily integrable leading to polynomial solutions. Three more differential equations described by (7) will augment systems (12) and (20) to solve for the rotational part. Boundary conditions are imposed at times 0 and 1. In the relaxation method, the ODE's are replaced by finite difference equations on a mesh of points corresponding to the time domain. If we define

$$x_1 = \sigma,\ x_2 = \omega,\ x_3 = \dot{\omega},\ x_4 = \ddot{\omega},$$

the geodesic (subscript g) and the minimum acceleration (subscript a) ODE's can be written as

$$\dot{x}_i = f_i(x_i),\ i \in \{g, a\} \quad (22)$$

where

$$x_g = [x_1^T\ x_2^T]^T,\ \ x_a = [x_1^T\ x_2^T\ x_3^T\ x_4^T]^T$$
$$f_g(x_g) = [f_1^T\ f_{g_2}^T]^T,\ \ f_a(x_a) = [f_1^T\ f_{a_2}^T\ f_{a_3}^T\ f_{a_4}^T]^T$$
$$f_1 = \left(I + \frac{1}{2}\hat{x}_1 + (1 - \alpha(\|x_1\|))\, \frac{\hat{x}_1^2}{\|x_1\|^2} \right) x_2 \quad (23)$$
$$f_{g_2} = -G^{-1}(x_2 \times (Gx_2)),\ f_{a_2} = x_3,\ f_{a_3} = x_4,\ f_{a_4} = -x_2 \times x_4$$

Note that the expression of f_1 is in accordance with the time derivative of exponential coordinates on $SO(3)$ as in (7) where $\alpha(y) = (y/2)cot(y/2)$. If N is the number of coupled first order differential equations in (22) and M is the number of mesh points, a solution consists of values for N dependent functions at each of the M mesh points, i.e. $N \times M$ variables in all. The relaxation method determines the solution by starting with a guess and improving it iteratively. At each step, the method produces a matrix equation whose solution is increments for each entry in the $N \times M$ matrix of interest. When this increments are sufficiently small, we say that the method has converged, or the matrix has relaxed to the true solution. For more details the interested reader is referred to [18].

If (23) is used, the relaxation method does not converge. The Jacobian of f_1 is very involved and the error propagation becomes significant. An approximation of (23) is used in the actual implementation, which is much simpler, although accurate for angular displacements σ in a (rather large) neighborhood of the origin. Equation (7) can be written equivalently as:

$$\dot{\sigma} = \left(I + \frac{1}{2}\hat{\sigma} + \sum_{n=1}^{\infty} \frac{B_{2n}}{(2n)!}\hat{\sigma}^{2n} \right) \omega \tag{24}$$

where B_i are the Bernoulli numbers ($B_0 = 1$, $B_1 = -\frac{1}{2}$, $B_2 = \frac{1}{6}$, $B_4 = \frac{1}{30}$,...). $B_n/n!$ are the coefficients of the Taylor expansion of the function $x/(e^x - 1)$ and therefore form a sequence convergent to zero. Limiting σ to a neighborhood of zero in $\mathbb{R}^3$, it makes sense to replace (24) with

$$\dot{\sigma} = \left(I + \frac{1}{2}\hat{\sigma} + \frac{1}{12}\hat{\sigma}^2 + \frac{1}{720}\hat{\sigma}^4 \right) \omega \tag{25}$$

and, therefore, (23) with

$$f_1 = \left(I + \frac{1}{2}\hat{x}_1 + \frac{1}{12}\hat{x}_1^2 + \frac{1}{720}\hat{x}_1^4 \right) x_2$$

4.2 Projection Method

The idea of the projection method [14] is to generate optimal curves in some ambient space with a constant metric and project the obtained trajectories on $SE(3)$.

We consider $SE(3)$ to be a submanifold (and a subgroup) of $GA(3)$ (the Lie group of affine maps in $\mathbb{R}^3$). The method involves two key steps: (1) the generation of optimal trajectories in $GA(3)$; and (2) the projection of the trajectories from $GA(3)$ to $SE(3)$. Due to the fact that the metric we define in $GA(3)$ is the same at all points, the corresponding Christoffel symbols [15] are all zero. Consequently, the optimal curves in the ambient space assume simple analytical forms (i.e geodesics - straight lines, minimum acceleration curves - cubic polynomial curves, minimum jerk curves - fifth order polynomial curves, all parameterized by time). The overall procedure is invariant with respect to both the local coordinates on the manifold and the choice of the inertial frame. The benefits of the method are three-fold. First, it is possible to apply any of the variety of well-known, efficient techniques to generate optimal curves on $GA(3)$ [1, 19]. Second, the method yields solutions for general choices of Riemannian metrics on $SE(3)$. Third, from a computational point of view, the method is less expensive than traditional methods, as shown in Section 6. The interested reader is referred to [14] for details.

An approximate geodesic for metric (8) (G is any rigid body inertia matrix) with boundary conditions given by (14) at $t = 0$ and $t = 1$ respectively, is given by (15) where

$$d(t) = d(0) + (d(1) - d(0))t, \quad R(t) = U(t)V^T(t)$$

with U and V determined from

$$M(t) = R(0) + (R(1) - R(0))t, \; M(t)W = U(t)\Sigma(t)V^T(t), \; W = \frac{1}{2}Tr(G)I_3 - G$$

An approximate minimum acceleration curve for the same position end conditions and velocity boundary conditions $\dot{R}(0)$, $\dot{d}(0)$, $\dot{R}(1)$, $\dot{d}(1)$ is given by (15) with

$$d(t) = d_0 + d_1 t + d_2 t^2 + d_3 t^3, \quad R(t) = U(t)V^T(t)$$

where

$$d_0 = d(0),\ d_1 = \dot{d}(0),\ d_2 = -3d(0) + 3d(1) - 2\dot{d}(0) - \dot{d}(1),\ d_3 = 2d(0) - 2d(1) + \dot{d}(0) + \dot{d}(1).$$

$$M(t) = M_0 + M_1 t + M_2 t^2 + M_3 t^3,$$

$$M_0 = R(0),\ M_1 = \dot{R}(0),\ M_2 = -3R(0) + 3R(1) - 2\dot{R}(0) - \dot{R}(1),\ M_3 = 2R(0) - 2R(1) + \dot{R}(0) + \dot{R}(1).$$

$$M(t)W = U(t)\Sigma(t)V^T(t),\ W = \frac{1}{2}Tr(G)I_3 - G$$

5 Simulation Results

A homogeneous parallelepipedic rigid body is assumed to move (rotate and translate) in free space. We assume that the body frame $\{M\}$ is placed at the center of mass and aligned with the principal axes of the body. Let a, b and c be the lengths of the body along its x, y and z axes respectively, and m the mass of the body.

For visualization, a small square is drawn on one of its faces and the center of the parallelepiped is shown starred.

The matrix G of metric $<,>_{SO}$ is given by

$$G = \begin{bmatrix} \frac{m}{24}(b^2+c^2) & 0 & 0 \\ 0 & \frac{m}{24}(a^2+c^2) & 0 \\ 0 & 0 & \frac{m}{24}(a^2+b^2) \end{bmatrix}$$

(a) (b)

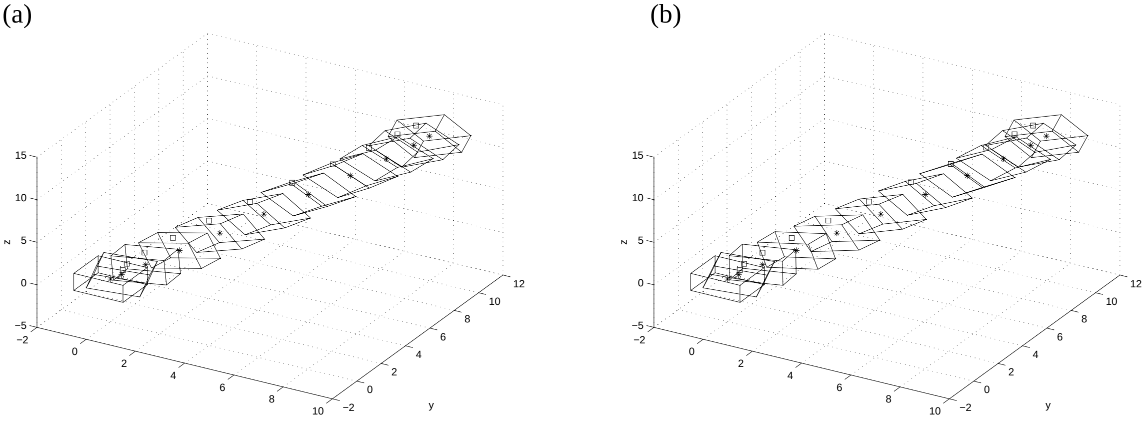

Figure 2: Minimum acceleration motion for a cube in free space: (a) relaxation method; (b) projection method.

The following boundary conditions were considered:

$$\sigma(0) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T,\ \sigma(1) = \begin{bmatrix} \frac{\pi}{6} & \frac{\pi}{3} & \frac{\pi}{2} \end{bmatrix}^T$$

$$\omega(0) = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}^T,\ \omega(1) = \begin{bmatrix} 2 & 1 & 1 \end{bmatrix}^T$$

$$d(0) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T,\ d(1) = \begin{bmatrix} 8 & 10 & 12 \end{bmatrix}^T$$

$$\dot{d}(0) = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T,\ \dot{d}(1) = \begin{bmatrix} 1 & 5 & 3 \end{bmatrix}^T$$

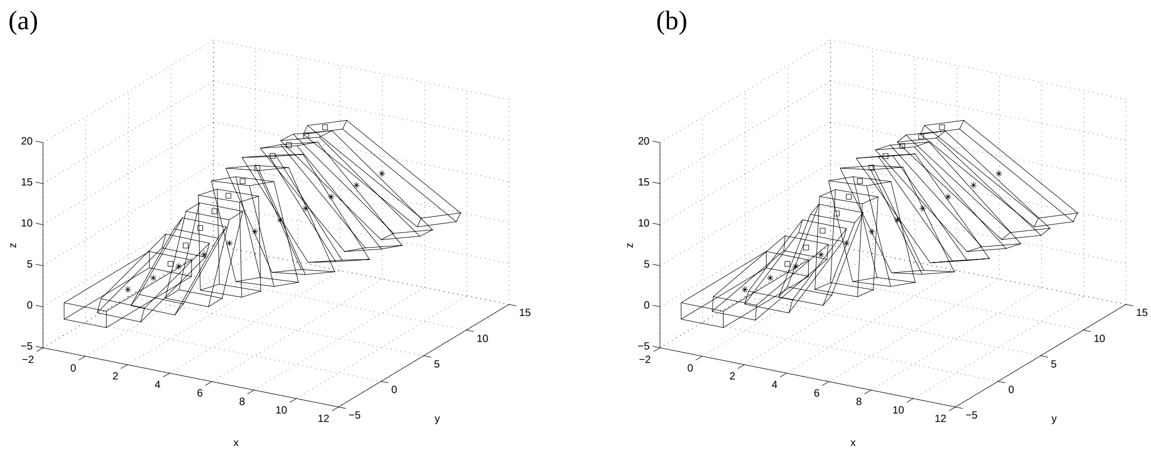

Figure 3: Geodesics for a parallelepipedic body: (a) relaxation method; (b) projection method.

True and projected minimum acceleration motions for a cubic rigid body with $a = b = c = 2$ and $m = 12$ are given in Figure 2 for comparison. Note that for this case $G = \alpha I$ with $\alpha = 4$.

Geodesics for the same boundary conditions and a parallelepipedic body with $a = c = 2$, $b = 10$ and $m = 12$ are given in Figure 3. For this case,

$$G = \begin{bmatrix} 52 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 52 \end{bmatrix}$$

As seen in Figures 2 and 3, even though the total displacement between the initial and final positions on $SO(3)$ is large (rotation angle of $\pi\sqrt{14}/6$), there is no noticeable difference between the true and the projected motions.

Three frames from the geodesic motion of the same parallelepiped generated using the relaxation method and Jack™ [1] are shown for illustration in Figure 4. Jack™ is a program which facilitates constructing geometric objects, positioning figures in a scene, and describing motion of the figures. It also has facilities for specifying lighting and surface property information.

6 Implementation and Computational Efficiency

It is not difficult to see that, from a computational point of view, it is less expensive to generate interpolating motion using the projection method as opposed to the relaxation method. Recall that the complexity of the

[1] Jack™ is a trademark of the University of Pennsylvania

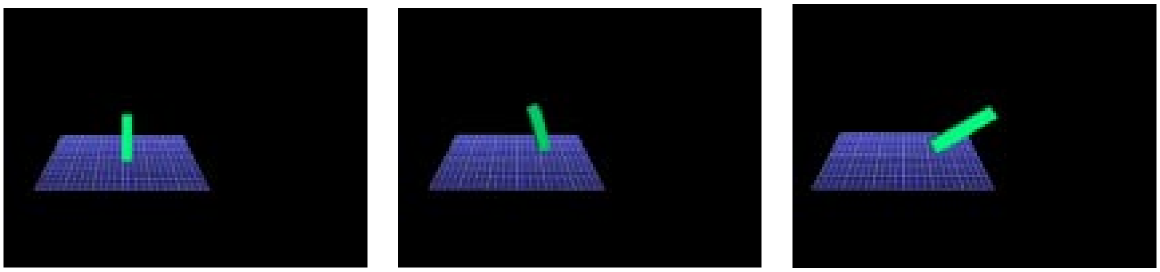

Figure 4: Frames from a movie produced with Jack™

SVD of a $n \times n$ matrix is of order n^3 [20]. If M is the number of uniformly distributed points in $[0,\, 1]$, then the number of flops required by the projection method in $GL(n)$ is of order $O(n^3 M)$

The relaxation method for generating solution at M mesh points of a system of N differential equations with two boundary conditions implies solving a $MN \times MN$ linear system in the corrections iteratively until the method relaxes to the solution (corrections converge to zero) [18]. Gaussian elimination, whose complexity is cubic, is used to solve the linear systems. Therefore, the number of flops required in the relaxation method is of order $O(M^3 N^3)$.

Consider the problem of generating geodesics on $SO(n)$. Here $N = n(n-1)$. The projection method involves $O(n^3 M)$ flops while the relaxation method has complexity of the order $O(n^6 M^3)$. For $M = 100$, as we used in this paper, the generation of geodesics on $SO(3)$ ($n = 3$) requires millions of flops by the relaxation method, while only thousands by the projection method.

The relaxation method has been programmed in C, using routines from [18]. The produced executable file is called with line command arguments from a MATLAB™ [2] file, so that the C code is transparent for the user. For both the projection and the relaxation method, the MATLAB™ files ask for input data: position and velocity boundary conditions, and mass and moments of inertia of the body. The output is the trajectory on $SE(3)$. The rotational part can be plotted separately in exponential coordinates. Files of type `.frames` to generate motion in Jack™ are also created.

7 Conclusion

This paper presents two methods for generating trajectories that minimize energy functionals on $SE(3)$, the group of rigid body displacements. The first method numerically integrates the Euler-Lagrange equations using a relaxation procedure. The second method gives an approximate but much simpler solution of the optimization problem. The two methods are compared and illustrative simulations are included.

References

[1] Farin, G. E., 1992. *Curves and surfaces for computer aided geometric design : a practical guide*. Academic Press, Boston, 3 edition.

[2] Hoschek, J., and Lasser, D., 1993. *Fundamentals of Computer Aided Geometric Design*. AK Peters.

[3] Shoemake, K., 1985. Animating rotation with quaternion curves. *ACM Siggraph*, 19(3):245–254.

[4] Ge, Q. J., and Ravani, B., 1994. Geometric construction of Bezier motions. *ASME Journal of Mechanical Design*, 116:749–755.

[2] MATLAB™ is a trademark of the MathWorks, Inc.

[5] Park, F. C., and Ravani, B., 1995. Bezier curves on Riemannian manifolds and Lie groups with kinematics applications. *ASME Journal of Mechanical Design*, 117(1):36–40.

[6] Ge, Q. J., and Ravani, B., 1994. Computer aided geometric design of motion interpolants. *ASME Journal of Mechanical Design*, 116:756–762.

[7] Jütler, B., 1994. Visualization of moving objects using dual quaternion curves. *Computers and graphics*, 18(3):315–326.

[8] Park, F. C., and Kang, I. G., 1996. Cubic interpolation on the rotation group using Cayley parameters. In *Proceedings of the ASME 24th Biennial Mechanisms Conference*, Irvine, CA.

[9] Noakes, L., Heinzinger, G., and Paden, B., 1989. Cubic splines on curved spaces. *IMA J. of Math. Control & Information*, 6:465–473.

[10] Crouch, P., and Silva Leite, F., 1995. The dynamic interpolation problem: on Riemannian manifolds, Lie groups, and symmetric spaces. *J. Dynam. Control Systems*, 1(2):177–202.

[11] Camarinha, M., Silva Leite, F., and Crouch, P., 1995. Splines of class c^k on non-Euclidean spaces. *IMA J. Math. Control Inform.*, 12(4):399–410.

[12] Žefran, M., Kumar, V., and Croke, C., 1995. On the generation of smooth three-dimensional rigid body motions. *IEEE Transactions on Robotics and Automation*, 14(4):579–589.

[13] Žefran, M., and Kumar, V., 1998. Interpolation schemes for rigid body motions. *Computer-Aided Design*, 30(3).

[14] Belta, C., and Kumar, V., 2000. New metrics for rigid body motion interpolation. In *Ball 2000 Symposium Commemorating the Legacy, Works, and Life of Sir Robert Stawell Ball*, University of Cambridge, UK.

[15] do Carmo, M. P., 1992. *Riemannian geometry*. Birkhauser, Boston.

[16] Murray, R. M., Li, Z., and Sastry, S. S., 1994. *A Mathematical Introduction to Robotic Manipulation*. CRC Press.

[17] Bullo, F., and Murray, R., 1995. Proportional derivative (pd) control on the Euclidean group. In *1995 European Control Conference*, Rome, Italy.

[18] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., 1988. *Numerical Recipes in C*. Cambridge University Press, Cambridge.

[19] Gallier, J., 2000. *Curves and Surfaces in Geometric Modeling - Theory and Algorithms*. Morgan Kaufmann Publishers, San Francisco, CA.

[20] Golub, G. H., and van Loan, C. F., 1989. *Matrix computations*. The Johns Hopkins University Press, Baltimore.