JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 23, No. 1, January–February 2000

Geometric Descent Algorithms for Attitude Determination Using the Global Positioning System

F. C. Park,* Junggon Kim,† and Changdon Kee‡
Seoul National University, Seoul 151-742, Republic of Korea

This paper describes a set of numerical gradient-based optimization algorithms for solving the Global Positioning System (GPS)-based attitude determination problem. We pose the problem as one of minimizing the function $tr(\Theta N \Theta^T Q - 2\Theta W)$ with respect to the rotation matrix Θ, where N, Q, and W are given 3×3 matrices, and $tr(\cdot)$ denotes the matrix trace. Both the method of steepest descent and Newton's method are generalized to the rotation group by taking advantage of its underlying Lie group structure. Analytic solutions to the line search procedure are also derived. Results of numerical experiments for the class of geometric descent algorithms proposed here are presented and compared with those of traditional vector space-based constrained optimization algorithms.

Introduction

THE Global Positioning System (GPS) has now entered the mainstream of engineering and is beginning to find applications ranging from land and air traffic control to surveying and aircraft precision landing. In this paper we address the problem of attitude determination using GPS and multiple antennas. Assuming that cycle ambiguities are already known, the problem can be posed as the minimization of a function on the rotation group $SO(3)$ of the form

$$J(\Theta) = tr(\Theta N \Theta^T Q - 2\Theta W) \tag{1}$$

where $\Theta \in SO(3)$ and N, Q, W are given 3×3 matrices. We show next how the usual formulation of the GPS-based attitude determination problem as given in the literature can be reduced to this particular form, and we describe the physical meaning of the elements constituting the various matrices. See Ref. 1 for a more complete discussion and background on the various issues related to GPS-based attitude determination.

The GPS-based attitude determination problem can be viewed as an optimization problem on $SO(3)$, and as is well-known, $SO(3)$ does not possess a Euclidean structure. There is both a well-developed theory and an abundance of numerical algorithms for constrained minimization of functions in Euclidean space, and it is quite reasonable to apply such an algorithm to the preceding objective function; the set of admissible solutions in this case would be the 3×3 real matrices Θ subject to the constraints $\Theta^T \Theta = I$ and $\det(\Theta) = 1$.

In this paper we suggest an alternative approach that takes advantage of the geometric structure of the rotation group $SO(3)$ and recasts the problem as one of unconstrained minimization. More generally, in the case of compact semisimple Lie groups like $SO(3)$, it is possible to exploit the geometric and algebraic structure of the underlying space and derive computationally efficient generalizations of unconstrained optimization algorithms that were originally intended for Euclidean space. In certain cases, and we show this to be true for our attitude determination problem, the geometric approach leads to algorithms with faster and more robust convergence properties.

Various forms of the attitude determination problem have been investigated in the aerospace literature. Closely related to our problem is the Wahba problem,[2] which can be posed as the minimization of the function

$$J(\Theta) = tr(\Theta F) \tag{2}$$

where $F \in \Re^{3\times 3}$ is a given an arbitrary nonsingular matrix constructed from sensor measurements. Observe that setting $N = I$ in our original objective function results in the Wahba problem (as does the trivial case of setting either N or Q to zero). Although not widely known within the aerospace community, the Wahba problem also arises in the context of the optimal matching of two three-dimensional point sets.[3–6] It can be shown (e.g., Refs. 7–9) that the Θ which minimizes and maximizes $J(\Theta)$ are respectively given by

$$\Theta_{\max} = \begin{cases} V\ \mathrm{diag}(1, 1, 1)U^T, & \det(F) > 0 \\ V\ \mathrm{diag}(1, 1, -1)U^T, & \det(F) < 0 \end{cases} \tag{3}$$

$$\Theta_{\min} = \begin{cases} V\ \mathrm{diag}(-1, -1, 1)U^T, & \det(F) > 0 \\ V\ \mathrm{diag}(-1, -1, -1)U^T, & \det(F) < 0 \end{cases} \tag{4}$$

where U and V are obtained from the singular value decomposition (SVD) of F, i.e., $F = U\Sigma V^T$ (see Ref. 10 for a discussion on how to choose U and V so that both $\Theta_{\max}$ and $\Theta_{\min}$ are proper rotation matrices).

Under certain simplifying assumptions, the GPS-based attitude determination problem reduces to the preceding form, but for the general case it does not appear that a closed-form solution like that for the Wahba problem can be derived. Numerical algorithms for the solution of the Wahba problem have also been proposed in, e.g., Refs. 7, 11–14, and in Ref. 15 a different optimality criterion for attitude estimation is proposed and evaluated. For the general GPS-based attitude determination problem, Cohen[1] develops a recursive algorithm for online estimation of the attitude that is based on a linearization of the objective function. Axelrad and Ward[16] present an extended Kalman filter for GPS-based attitude determination expressed in terms of quaternions, whereas Axelrad and Behre[17] discuss and compare several time and frequency algorithms for estimating spinning spacecraft angular rates and orientation of the angular momentum vector using GPS phase data. More recently Bar-Itzhack et al.[18] have proposed iterative quaternion-based algorithms for GPS-based attitude determination.

Although not directed specifically at GPS-based attitude determination, in related work a number of researchers have posed optimization problems on $SO(n)$ and investigated their solutions in a geometric setting. Brockett[19] considers objective functions on $SO(n)$ of the form

$$J(\Theta) = tr(\Theta N \Theta^T Q) \tag{5}$$

where N, Q are given symmetric $n \times n$ matrices. He shows that a variety of continuous and combinatorial problems (e.g., the eigenvalue

Received 10 August 1998; revision received 9 July 1999; accepted for publication 10 July 1999.

*Associate Professor, School of Mechanical and Aerospace Engineering.
†Graduate Student, School of Mechanical and Aerospace Engineering.
‡Assistant Professor, School of Mechanical and Aerospace Engineering. Member AIAA.

problem, linear programming, sorting) can be posed as the minimization of such an objective function and derives the corresponding gradient flow equations. Chu,[20] Bloch et al.,[21] Helmke and Moore,[22] and others (see, e.g., Brockett[23] and the references cited therein) examine the geometry of these and other related optimization problems on non-Euclidean spaces.

In this article we derive a class of numerical optimization algorithms for GPS-based attitude determination that takes advantage of the special structure of $SO(3)$. Specifically, we present geometric versions of steepest descent and Newton's method applicable to our particular objective function on $SO(3)$; in both these methods the attitude is determined as the solution of an unconstrained minimization problem. After some geometric preliminaries we derive the detailed steps of the geometric steepest descent algorithm, including methods for finding the exact solution as well as rapid estimates of the step size in the line search procedure. Newton's method is then generalized to our objective function. Results of numerical experiments comparing the relative performance of these algorithms with standard minimization algorithms provided by MATLAB® are also given.

The particular problem we address ignores a number of practical issues related to GPS systems that must be resolved in order to obtain a complete solution, e.g., cycle ambiguity resolution, the multipath problem, structural distortion caused by flexure and thermal effects, tropospheric effects, and receiver-specific errors, all of which are topics of current study. Nevertheless, the problem we address is central to any attitude determination that uses GPS, and we believe our results suggest a fundamentally different and promising approach to high-performance attitude determination.

Geometric Preliminaries

In this section we review the geometric structure of the rotation group $SO(3)$ and formulate the GPS-based attitude determination problem as the minimization of a certain second-order function on $SO(3)$. Some familiarity with matrix groups at the level of Ref. 24 is helpful but not required. The survey[12] also provides a detailed treatment of $SO(3)$ and its various representations.

Exponential Coordinates on *SO*(3)

Recall that $SO(3)$ consists of all 3×3 real matrices Θ satisfying $\Theta^T\Theta = I$ and $\det\Theta = 1$. $SO(3)$ has the structure of both an algebraic group and a differentiable manifold and is the classical example of a Lie group (see, e.g., Refs. 25 and 26 for an in-depth discussion). The Lie algebra of $SO(3)$, denoted in lower-case as $so(3)$, consists of the 3×3 real skew-symmetric matrices, i.e.,

$$so(3) = \{S \in \Re^{3\times 3} \mid S + S^T = 0\} \tag{6}$$

with the Lie bracket $[\cdot, \cdot]$: $so(3) \times so(3) \to so(3)$ defined by the matrix commutator:

$$[S_1, S_2] = S_1S_2 - S_2S_1 \tag{7}$$

We introduce the following piece of notation. Given a vector $\omega = (\omega_1, \omega_2, \omega_3)$ and a skew-symmetric matrix

$$S = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{8}$$

then

$$[\omega] = S \tag{9}$$

$$\boldsymbol{S} = \omega \tag{10}$$

Observe that $[S_1, S_2] = [\boldsymbol{S}_1 \times \boldsymbol{S}_2]$, i.e., the Lie bracket on $so(3)$ corresponds to the vector product in $\Re^3$.

As with other Lie groups, the exponential map exp: $so(3) \to SO(3)$ provides a set of local coordinates for $SO(3)$. If $\omega \in \Re^3$, $\|\omega\| = 1$, and $\theta \in \Re$ is an arbitrary scalar, then it is well-known that

$$\exp([\omega]\theta) = I + \sin\theta[\omega] + (1 - \cos\theta)[\omega]^2 \tag{11}$$

is an onto map with the following physical interpretation: $\exp([\omega]\theta)$ is a rotation matrix corresponding to a rotation about the axis ω by an angle θ (in the right-hand sense). Conversely, given $\Theta \in SO(3)$ such that $tr(\Theta) \neq -1$,

$$\log\Theta = (\phi/2\sin\phi)(\Theta - \Theta^T)$$

where ϕ satisfies $1 + 2\cos\phi = tr(\Theta)$, $|\phi| < \pi$, and $\|\log\Theta\|^2 = \phi^2$.

The preceding formulas suggest the standard visualization of $SO(3)$ as a ball of radius π, centered at the origin with the antipodal points identified—a point p inside the ball represents a rotation by an angle of $\|p\|$ radians about the line passing from the origin through p. The rotation matrices with trace -1 have a rotation angle of π, and their logarithms are points on the boundary of the solid ball. In this case $\log\Theta$ can have two possible values: if v is a unit eigenvector of Θ associated with the eigenvalue 1, then $\log\Theta = \pm\pi[v]$. As will be familiar to many, the exponential and logarithm are simply precise mathematical descriptions of the Euler angle and Euler axis representation for rotations.

We define one additional mapping, called the adjoint mapping: if X is an element of a Lie group and x an element of its corresponding Lie algebra, then $Ad_X(x) = XxX^{-1}$ is an element of the Lie algebra. The following properties of the adjoint mapping on $SO(3)$ can be verified via direct calculation and are used in what follows:

1) $\Theta[\omega]\Theta^T = [\Theta\omega]$ for arbitrary $\Theta \in SO(3)$ and $\omega \in \Re^3$.

2) $Ad_{\Theta_1}Ad_{\Theta_2} = Ad_{\Theta_1\Theta_2}$ for arbitrary $\Theta_1, \Theta_2 \in SO(3)$.

3) $Ad_\Theta([\omega_1, \omega_2]) = [Ad_\Theta(\omega_1), Ad_\Theta(\omega_2)]$ for arbitrary $\Theta \in SO(3)$ and $\omega_1, \omega_2 \in so(3)$.

The last two identities are also valid for general Lie groups.

Minimal Geodesics on *SO*(3)

A natural way to define lengths of curves on $SO(3)$ is as follows. Let $\Theta(t)$ be a differentiable curve on $SO(3)$ defined over the closed interval $0 \leq t \leq 1$. It is easily verified that both $\Theta^T\dot{\Theta} = \Omega_b$ and $\dot{\Theta}\Theta^T = \Omega_s$ are elements of $so(3)$. Physically $\boldsymbol{\Omega}_b$ and $\boldsymbol{\Omega}_s$ correspond to the angular velocity expressed in body and inertial frame coordinates, respectively. Define the inner product $\langle\cdot, \cdot\rangle$ on $so(3)$ by

$$\langle\Omega_1, \Omega_2\rangle = tr(\Omega_1^T\Omega_2) = 2\boldsymbol{\Omega}_1^T\boldsymbol{\Omega}_2 \tag{12}$$

This inner product can be extended to the tangent space at an arbitrary point Θ of $SO(3)$ by either left or right translation. In the case of left translation,

$$\langle\dot{\Theta}, \dot{\Theta}\rangle = \langle\Omega_b, \Omega_b\rangle \tag{13}$$

From the results of the preceding section, one can see that $\langle\Omega_s, \Omega_s\rangle = \langle\Omega_b, \Omega_b\rangle$. Our particular choice of inner product on $so(3)$ therefore leads to an inner product on the tangent space of $SO(3)$ that is both left and right invariant, or bi-invariant.

The length of the curve $\Theta(t)$ is then defined as

$$\mathcal{L} = \int_0^1 \langle\dot{\Theta}, \dot{\Theta}\rangle^{\frac{1}{2}}\,dt \tag{14}$$

This definition of length is natural in the sense that it is invariant with respect to both left and right translations, i.e., the length of both $R\Theta(t)$ and $\Theta(t)R$ for arbitrary constant $R \in SO(3)$ is the same as that of $\Theta(t)$. Given arbitrary endpoints $\Theta_0, \Theta_1 \in SO(3)$, the minimum length curve $\Theta(t)$ connecting these endpoints (referred to as the minimal geodesic) is given by

$$\Theta(t) = \Theta_0\exp(At), \qquad A = \log\left(\Theta_0^T\Theta_1\right) \tag{15}$$

Note that the minimal geodesic corresponds to the minimum angle eigenaxis rotation between two orientations. The exponential coordinates provide a convenient and mathematically concise means of representing and evaluating minimal geodesics on $SO(3)$. Moreover, the minimal geodesic allows one to extend the notion of straight lines to curved spaces like $SO(3)$ and provides the means for generalizing the steepest descent method to minimize functions on $SO(3)$, as we show later.

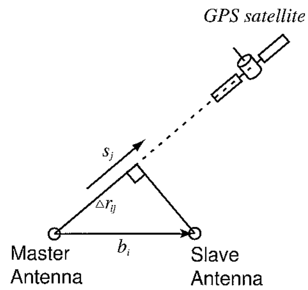

Fig. 1 Physical meaning of $\triangle R_{ij}$, b_i, and s_j.

GPS-Based Attitude Determination: Problem Formulation

The standard formulation of the GPS-based attitude determination problem is to minimize the following objective function with respect to a rotation matrix Θ:

$$\min_{\Theta \in SO(3)} \sum_{i=1}^{m} \sum_{j=1}^{n} (\triangle r_{ij} - b_i^T \Theta s_j)^2 \tag{16}$$

where $\triangle r_{ij}$ denotes the set of differential range measurements taken at a single epoch for baseline i and satellite j, m, and n respectively denote the number of baselines and satellites, $b_i \in \Re^3$ is the ith baseline vector defined in the body frame, and $s_j \in \Re^3$ is the line of sight to the jth satellite given in the local horizontal frame (see Fig. 1 and Ref. 1). Θ, the optimization parameter, represents the orientation of the body frame relative to the local horizontal frame. Defining $R = (\triangle r_{ij}) \in \Re^{m \times n}$, $B = (b_1, \ldots, b_m) \in \Re^{3 \times m}$, $S = (s_1, \ldots, s_n) \in \Re^{3 \times n}$, the preceding objective function can be recast into the following form:

$$\begin{aligned} \sum_{i=1}^{m} \sum_{j=1}^{n} (\triangle r_{ij} - b_i^T \Theta s_j)^2 &= \|R - B^T \Theta S\|^2 \\ &= tr\{(R - B^T\Theta S)(R - B^T \Theta S)^T\} \\ &= tr\{B^T \Theta S S^T \Theta^T B - 2B^T \Theta S R^T + RR^T\} \\ &= tr\{\Theta N \Theta^T Q - 2\Theta W\} + tr\{RR^T\} \end{aligned} \tag{17}$$

where $N = SS^T$, $Q = BB^T$, and $W = SR^T B^T$, and we have made use of the general identity $tr(ABC) = tr(CAB) = tr(BCA)$. Because the last term is constant, it can be neglected, so that the objective function reduces to

$$\min_{\Theta \in SO(3)} tr\{\Theta N \Theta^T Q - 2\Theta W\} \tag{18}$$

Observe that the objective function consists of the sum of a linear and square term in Θ, with N, Q symmetric by definition, and W arbitrary.

In Ref. 27 an even more general objective function is formulated by introducing a weighting factor w_{ij}:

$$\min_{\Theta \in SO(3)} \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} (\triangle r_{ij} - b_i^T \Theta s_j)^2 \tag{19}$$

The net effect of introducing the weighting factor is that, although the final form of the objective function is still $tr\{\Theta N \Theta^T Q - 2\Theta W\}$, the matrices N and Q are no longer required to be symmetric. We explain the main ideas of our algorithm for the symmetric case and separately address the asymmetric case, which requires only minor modification of the symmetric results.

Steepest Descent Algorithm

The steepest descent method is one of the most basic and fundamental algorithms in numerical optimization and serves as the basis for more sophisticated optimization algorithms. Given an objective function $J(x)$ in Euclidean space, the steepest descent algorithm consists of the following steps:

1) Find the gradient of the objective function at the present point.
2) Find the point on the line in the gradient direction that minimizes the objective function (line search).
3) Repeat the preceding steps until convergence occurs.

Mathematically, the iteration can be expressed as

$$x_{k+1} = x_k - \alpha_k \nabla J(x_k) \tag{20}$$

where

$$\alpha_k = \arg\min_{\alpha} J[x_k - \alpha \nabla J(x_k)] \tag{21}$$

A natural means of extending the steepest descent algorithm to functions on $SO(3)$ is to conduct the line search along the minimal geodesic in the direction of the gradient. In the next section we derive the analytic gradient for our particular objective function and discuss methods for performing an efficient line search along the minimal geodesic.

Computing the Gradient

The gradient can be derived by examining the first-order term in the series expansion of the objective function. We parameterize a neighborhood of $\Theta_0 \in SO(3)$ by

$$\Theta(\Omega) = \Theta_0(I + \Omega + \Omega^2/2! + \cdots) \tag{22}$$

with $\Omega \in so(3)$. Then to first order in Ω, we have

$$\begin{aligned} &tr\left\{\Theta_0(I + \Omega)N(I - \Omega)\Theta_0^T Q - 2\Theta_0(I + \Omega)W\right\} \\ &= tr\left\{\Theta_0 N \Theta_0^T Q - 2\Theta_0 W\right\} + tr\left\{\Theta_0 \Omega N \Theta_0^T Q \right. \\ &\left. - \Theta_0 N \Omega \Theta_0^T Q - 2\Theta_0 \Omega W\right\} + o(\Omega) \end{aligned} \tag{23}$$

Ignoring higher-order terms, and taking advantage of the fact that N and Q are symmetric (for now—we treat the asymmetric case separately in the following), the term linear in Ω can be simplified to

$$\langle (N\Theta_0^T Q \Theta_0 - W\Theta_0) - (N\Theta_0^T Q\Theta_0 - W\Theta_0)^T, \Omega \rangle \tag{24}$$

where we have also used the easily established fact that

$$tr(A\Omega) = \tfrac{1}{2} tr\{(A - A^T)\Omega\} \tag{25}$$

for any $A \in \Re^{n \times n}$ and $\Omega \in so(n)$ (i.e., $\Omega + \Omega^T = 0$). The gradient of the objective function at Θ is therefore represented by

$$\langle (N\Theta^T Q\Theta - W\Theta) - (N\Theta^T Q\Theta - W\Theta)^T, \cdot \rangle \tag{26}$$

and the minimal geodesic passing through Θ_0 in the direction of the gradient is given by

$$\Theta(t) = \Theta_0 e^{\Omega t}$$
$$\Omega = (N\Theta_0^T Q\Theta_0 - W\Theta_0) - (N\Theta_0^T Q\Theta_0 - W\Theta_0)^T \tag{27}$$

We can also express the gradient flow using $\dot{\Theta} = \Omega\Theta$ as

$$\Theta^T \dot{\Theta} = (N\Theta^T Q\Theta - W\Theta) - (N\Theta^T Q\Theta - W\Theta)^T \tag{28}$$

or

$$\dot{\Theta} = \Theta N \Theta^T Q \Theta - Q\Theta N - \Theta W \Theta - W^T \tag{29}$$

The steepest descent algorithm is therefore given by the following iteration:

Steepest Descent Algorithm:

$$\Omega_k = (N\Theta_k^T Q\Theta_k - W\Theta_k) - (N\Theta_k^T Q\Theta_k - W\Theta_k)^T \tag{30}$$

$$t_k = \min_{t \in \Re} J(\Theta_k e^{\Omega_k t}) \tag{31}$$

$$\Theta_{k+1} = \Theta_k e^{\Omega_k t_k} \tag{32}$$

In the next section we discuss methods for performing the line search to find the stepsize t_k.

Line Search

In this section we present an exact solution to the line-search minimization problem, as well as an alternative formulation of Brockett's[23] method for determining a step-size bound that ensures a decrease in the objective function.

Exact Solution

In the case of $SO(3)$, because an explicit formula exists for the exponential coordinates, one can obtain the exact solution of the line-search procedure by simply determining the roots of a fourth-order polynomial. The method proceeds as follows. Recalling that $e^{\Omega t} = I + \Omega \sin t + \Omega^2(1 - \cos t)$ for unit length Ω, the line-search minimization procedure can be expressed as

$$\min_{t \in \Re} \phi(t) = tr\{\Theta e^{\Omega t} N e^{-\Omega t} \Theta^T Q - 2\Theta e^{\Omega t} W\} \tag{33}$$

Observe that $\phi(t)$ is periodic with period 2π. Substituting for $e^{\Omega t}$ and simplifying, $\phi(t)$ can be expressed as

$$\phi(t) = c_1 \sin t + c_2 \cos t + c_3 \sin 2t + c_4 \cos 2t + c_5 \tag{34}$$

with the coefficients defined as $c_1 = a + c$, $c_2 = -b - 2e$, $c_3 = -c/2$, $c_4 = -(d/2) + (e/2)$, and $c_5 = b + d/2 + 3e/2 + f$, where

$$a = tr\{(\Omega N - N\Omega)\Theta^T Q\Theta - 2\Omega W\Theta\}$$

$$b = tr\{(N\Omega^2 + \Omega^2 N)\Theta^T Q\Theta - 2\Omega^2 W\Theta\}$$

$$c = tr\{(\Omega N\Omega^2 - \Omega^2 N\Omega)\Theta^T Q\Theta\}$$

$$d = -tr\{\Omega N\Omega\Theta^T Q\Theta\}$$

$$e = tr\{\Omega^2 N\Omega^2\Theta^T Q\Theta\}$$

$$f = tr\{N\Theta^T Q\Theta - 2W\Theta\}$$

An alternative means of determining the coefficients that takes advantage of the periodicity of $\phi(t)$ is as follows:

$$\begin{array}{ll} g_0 = \phi(0) = c_2 + c_4 + c_5 & c_1 = \frac{1}{2}(g_1 + g_3) \\ g_1 = \phi(\pi/2) = c_1 - c_4 + c_5 & c_2 = \frac{1}{2}(g_0 - g_2) \\ g_2 = \phi(\pi) = -c_2 + c_4 + c_5 \quad \Rightarrow & c_3 = g_4 - [(\sqrt{2}+1)/4](g_0 + g_1) + [(\sqrt{2}-1)/4](g_2 + g_3) \\ g_3 = \phi(\frac{3}{2}\pi) = -c_1 - c_4 + c_5 & c_4 = \frac{1}{4}(g_0 - g_1 + g_2 - g_3) \\ g_4 = \phi(\pi/4) = c_1/\sqrt{2} + c_2/\sqrt{2} + c_3 + c_5 & c_5 = \frac{1}{4}(g_0 + g_1 + g_2 + g_3) \end{array} \tag{35}$$

The first-order necessary conditions for a local minimum are given by the roots of

$$\frac{d}{dt}\phi(t) = c_1 \cos t - c_2 \sin t + 2c_3 \cos 2t - 2c_4 \sin 2t = 0 \tag{36}$$

Making the trigonometric half-angle substitution $x = \tan(x/2)$, so that $\sin t = 2x/(1 + x^2)$, $\cos t = (1 - x^2)/(1 + x^2)$, $\sin 2t = [4x(1 - x^2)]/[(1 + x^2)^2]$, and $\cos 2t = (x^4 - 6x^2 + 1)/[(1 + x^2)^2]$, the first-order necessary conditions reduce to the following fourth-order polynomial in x:

$$(2c_3 - c_1)x^4 - 2(c_2 - 4c_4)x^3 - 12c_3x^2 - 2(c_2 + 4c_4)x + (c_1 + 2c_3) = 0 \tag{37}$$

The optimal step size is given by the real root of the preceding quartic (formulas for the roots of a quartic polynomial are available in various mathematical references) that minimizes the objective function ϕ.

Step-Size Estimate

In Ref. 23 Brockett derives an estimate for the step size in the steepest descent minimization of the function $tr(\Theta N\Theta^T Q)$. Specifically, he derives an estimate for the t that minimizes

$$\phi(t) = tr\{Ad_{\Theta e^{\Omega t}}(N)\}Q \tag{38}$$

for given Θ, Ω, N, Q. In this section we derive a similar step-size estimate for the objective function $tr(\Theta N\Theta^T Q - 2\Theta W)$. The basic idea rests on the observation that for a scalar differentiable function $f(t)$ such that $f(0) > 0$ and $|f'(t)| \leq c$ for all t, then $f(t) \geq 0$ in the interval $-f(0)/c \leq t \leq f(0)/c$. Similarly, if $f'(0) > 0$ and $|f''(t)| \leq c$ for all t, then $f'(t) \geq 0$ in the interval $-f'(0)/c \leq t \leq f'(0)/c$. If on the other hand we have $f'(0) < 0$ and $|f''(t)| \leq c$ for all t, then $f'(t) \leq 0$ in the interval $f'(0)/c \leq t \leq -f'(0)/c$.

An estimate for the step size can now be obtained by a combination of the preceding result and the Cauchy–Schwarz Inequality. Let $J(\Theta) = tr(\Theta N\Theta^T Q - 2\Theta W)$, and

$$\phi(t) = J(\Theta e^{\Omega t}) \tag{39}$$

$$= tr\{Ad_{\Theta e^{\Omega t}}(N)Q\} - 2tr(\Theta e^{\Omega t} W) \tag{40}$$

Then

$$\phi'(t) = tr\{Ad_{\Theta e^{\Omega t}}(ad_\Omega N)Q\} - 2tr(\Theta e^{\Omega t}\Omega W) \tag{41}$$

and

$$\phi''(t) = -tr\{Ad_{e^{\Omega t}}(ad_\Omega N)ad_\Omega[Ad_{\Theta^T}(Q)]\} - 2tr(\Theta e^{\Omega t}\Omega^2 W) \tag{42}$$

Because the adjoint mapping Ad on $SO(3)$ is an isometry, by applying the Cauchy–Schwarz Inequality to each term on the right-hand side we obtain

$$|\phi''(t)| \leq \|ad_\Omega(N)\| \cdot \|ad_\Omega[Ad_{\Theta^T}(Q)]\| + 2\sqrt{3}\|\Omega^2 W\| \tag{43}$$

Observing that $\phi'(0) = -2tr\{\Theta N\Omega\Theta Q + \Theta\Omega W\}$ and applying our earlier result if $\phi'(0) < 0$, then by choosing

$$t^* = \frac{|\phi'(0)|}{\|ad_\Omega N\| \cdot \|ad_\Omega[Ad_{\Theta^T}(Q)]\| + 2\sqrt{3}\|\Omega^2 W\|} \tag{44}$$

we can be assured $\phi(t^*) < \phi(0)$. Hence, at each step of the steepest descent algorithm the objective function is guaranteed to decrease. If $\phi'(0) > 0$, then setting t^* to have the opposite sign will accomplish the same result.

The step-size estimate is meaningful only for steepest descent; when used with other descent algorithms in which the search direction does not coincide with the gradient, convergence of the algorithm cannot be guaranteed. To understand why, observe that for general descent algorithms the search direction at each iteration will be of the form $d = Ag$, where g is the gradient direction and A is some suitable square matrix (e.g., for steepest descent, $A = -I$). The step-size estimate can be expressed in the form $t^* = \|\phi'(0)\|/c$, where c is some suitable constant greater than or equal to $\|\phi''(t)\|$. Algorithms using the step-size estimate will then converge to a value in which $\phi'(0) = 0$, or equivalently $g^d = g^T Ag = 0$. This result implies that either $g = 0$, which assures convergence to a local minimum, or when this is not the case, $g^T Ag = 0$ with $g \neq 0$. In the case of steepest descent, because A is always equal to $-I$ this corresponds to the former case and convergence to a local minimum can be guaranteed. However, for other descent algorithms in which A is not always positive definite (e.g., Newton's method, which we discuss below), convergence to a local minimum cannot in general be guaranteed.

Asymmetric Case

For the case of asymmetric N and Q, the steepest descent algorithm is given by the following iteration:

Steepest Descent Algorithm (Asymmetric Case):

$$\Omega_k = (N\Theta_k^T Q\Theta_k - \Theta_k^T Q\Theta_k N - 2W\Theta_k) - (N\Theta_k^T Q\Theta_k - \Theta_k^T Q\Theta_k N - 2W\Theta_k)^T \quad (45)$$

$$t_k = \min_{t\in\Re} J(\Theta_k e^{\Omega_k t}) \quad (46)$$

$$\Theta_{k+1} = \Theta_k e^{\Omega_k t_k} \quad (47)$$

Newton's Method

The method of steepest descent method has linear convergence, and one of its well-known disadvantages is that it converges extremely slowly near critical points where the Hessian of the objective function is poorly conditioned. Newton's method, which uses second-order terms in the Taylor expansion, in contrast has quadratic convergence. In this section we develop a Newton-type algorithm for our GPS-based attitude determination problem.

The idea behind Newton's method is that the function to be minimized is approximated locally by a quadratic function, and this approximated function is then minimized exactly. In the case of functions in Euclidean space, the function $J(x)$ is approximated near x_k by the second-order Taylor expansion

$$J(x) = J(x_k) + \nabla J(x_k)(x - x_k) + \tfrac{1}{2}(x - x_k)^T \nabla^2 J(x_k)(x - x_k) \quad (48)$$

where $\nabla^2 J(x) = \partial^2 J/\partial x^2$ is the Hessian of J at x. Solving the first-order necessary conditions and adding a step-size minimization procedure, Newton's algorithm can be expressed as

$$x_{k+1} = x_k - \alpha_k \left\{\nabla^2 J(x_k)\right\}^{-1} \nabla J(x_k)^T \quad (49)$$

where the step size α_k is chosen to minimize J along the search direction. Observe that replacing $\nabla^2 J(x_k)$ by the identity results in the steepest descent algorithm.

Symmetric Case

To generalize this method to the symmetric form of our objective function, the first order of business is to expand the objective function to second order about some Θ_k:

$$J(\Theta) = J(\Theta_k) + tr\left\{2(N\Theta_k^T Q\Theta_k - W\Theta_k)\Omega\right\} + \tfrac{1}{2} tr\left\{2(N\Theta_k^T Q\Theta_k - W\Theta_k)\Omega^2 - 2\Theta_k^T Q\Theta_k \Omega N \Omega\right\} + o(\Omega^2) \quad (50)$$

where again we use left-translated exponential coordinates $\Theta = \Theta_k(I + \Omega + \frac{1}{2}\Omega^2 + \cdots)$ for the expansion. The second-order Taylor expansion is of the form

$$J(\Theta) = J(\Theta_k) + tr(F\Omega) + \tfrac{1}{2} tr(A\Omega^2 + B\Omega C\Omega) + o(\Omega^2) \quad (51)$$

where F, A, B, C are given 3×3 matrices and $\Omega \in so(3)$. Newton's algorithm can be obtained by differentiating the preceding with respect to Ω and setting it equal to zero and finding the root Ω. The following results are useful in this regard.

Proposition 1: Given $F, A, B, C \in \Re^{3\times 3}$ and $\Omega \in so(3)$. Then,

$$\left[\frac{\partial}{\partial \mathbf{\Omega}} tr(F\Omega)\right] = F^T - F \quad (52)$$

$$\left[\frac{\partial}{\partial \mathbf{\Omega}} \frac{1}{2} tr(A\Omega^2)\right] = -(T\Omega + \Omega T) \quad (53)$$

$$\left[\frac{\partial}{\partial \mathbf{\Omega}} \frac{1}{2} tr(B\Omega C\Omega)\right] = -(M\Omega N + N\Omega M) \quad (54)$$

where $T = \frac{1}{2}(A + A^T)$, $M = \frac{1}{2}(B + B^T)$, $N = \frac{1}{2}(C + C^T)$. The preceding identities can be verified by a straightforward calculation.

Differentiating the terms up to second order in the expansion for $J(\Theta)$ and setting the result to zero, one obtains

$$S - T\Omega - \Omega T - M\Omega N - N\Omega M = 0 \quad (55)$$

where

$$S = 2\left\{(N\Theta_k^T Q\Theta_k)^T - N\Theta_k^T Q\Theta_k + \Theta_k^T W - W\Theta_k\right\} \quad (56)$$

$$T = N\Theta_k^T Q\Theta_k + (N\Theta_k^T Q\Theta_k)^T - W\Theta_k - \Theta_k^T W \quad (57)$$

$$M = -2\Theta_k^T Q\Theta_k \quad (58)$$

The root Ω can be solved in closed form as follows:

$$\mathbf{\Omega} = \{MN + NM - (tr\ M)N - (tr\ N)M + (tr\ M\ tr\ N - tr\ MN)I + (tr\ T)I - T\}^{-1}\mathbf{S} \quad (59)$$

The preceding equation can also be verified by a straightforward but involved calculation.

In summary, we can construct the following Newton-type iterative procedure for solving the GPS-based attitude determination problem:

Algorithm for Newton's Method:

$$T = N\Theta_k^T Q\Theta_k + (N\Theta_k^T Q\Theta_k)^T - W\Theta_k - \Theta_k^T W \quad (60)$$

$$M = -2\Theta_k^T Q\Theta_k \quad (61)$$

$$S = 2\left\{(N\Theta_k^T Q\Theta_k)^T - N\Theta_k^T Q\Theta_k + \Theta_k^T W - W\Theta_k\right\} \quad (62)$$

$$H = MN + NM - (tr\ M)N - (tr\ N)M + (tr\ M\ tr\ N - tr\ MN)I + (tr\ T)I - T \quad (63)$$

$$\mathbf{\Omega}_k = H^{-1}\mathbf{S} \quad (64)$$

$$t_k = \min_{t\in\Re} J(\Theta_k e^{\Omega_k t}) \quad (65)$$

$$\Theta_{k+1} = \Theta_k e^{\Omega_k t_k} \quad (66)$$

Asymmetric Case

The algorithm for asymmetric N and Q can be obtained by a straightforward modification of the results for the symmetric case. The resulting algorithm is as follows:

Algorithm for Newton's Method (Asymmetric Case):

$$T = \tfrac{1}{2}(N\Theta_k^T Q\Theta_k + \Theta_k^T Q\Theta_k N - 2W\Theta_k)^T + \tfrac{1}{2}(N\Theta_k^T Q\Theta_k + \Theta_k^T Q\Theta_k N - 2W\Theta_k) \quad (67)$$

$$M = -\Theta_k^T (Q + Q^T)\Theta_k \quad (68)$$

$$S = (N\Theta_k^T Q\Theta_k - \Theta_k^T Q\Theta_k N - 2W\Theta_k)^T - (N\Theta_k^T Q\Theta_k - \Theta_k^T Q\Theta_k N - 2W\Theta_k) \quad (69)$$

$$U = \tfrac{1}{2}(N + N^T) \quad (70)$$

$$H = MU + UM - (tr\ M)U - (tr\ U)M + (tr\ M\ tr\ U - tr\ MU)I + (tr\ T)I - T \quad (71)$$

$$\mathbf{\Omega}_k = H^{-1}\mathbf{S} \quad (72)$$

$$t_k = \min_{t\in\Re} J(\Theta_k e^{\Omega_k t}) \quad (73)$$

$$\Theta_{k+1} = \Theta_k e^{\Omega_k t_k} \quad (74)$$

Numerical Results

We now describe results of numerical experiments for evaluating the performance of the geometric algorithms considered in this paper. One rather obvious strategy that we have not yet discussed in this paper is to simply parametrize Θ by some suitable three-parameter representation (e.g., Euler angles, exponential coordinates) and solve the resulting unconstrained minimization problem. For comparison purposes we consider exponential coordinates, so that the objective function is of the form

$$J(x) = tr(e^{[x]} N e^{-[x]} Q - 2We^{[x]}) \tag{75}$$

where $\|x\| \leq \pi$. Other coordinate representations of $SO(3)$ are possible, e.g., Euler angles, Rodrigues parameters, unit quaternions (which involves four coordinates but introduces an additional constraint), etc. For our study we have not carried out a systematic analysis of the effects of the particular choice of local coordinates; some recent related work in this connection is that of Oshman and Markley.[28] However, preliminary experiments with these various representations suggest that the exponential coordinates result in the fastest and most robust convergence properties for the algorithms considered.

We apply two popular unconstrained minimization algorithms provided by MATLAB: the Nelder–Mead-type simplex search method and the BFGS quasi-Newton method [invoked by the commands fmins( ) and fminu( ), respectively]. Because of the symbolic complexity of the objective function, it is infeasible to determine analytic expressions for the gradient and Hessian in terms of local coordinates; these quantities are instead determined numerically by the MATLAB routines. Also, like general user-specified MATLAB programs, both fmins( ) and fminu( ) are written as m-files using MATLAB's programming language. For uniform comparison the geometric algorithms of this paper are therefore also implemented as m-files.

For all of the numerical experiments, the elements of N, Q, and W, as well as the initial point Θ_0, are generated using the random function provided by MATLAB; this generates a value between 0 and 1 of uniform probability (the effects of scaling the values of N, Q, and W are discussed next). The stopping criterion we use is $\|\Theta_{k+1} - \Theta_k\| < 10^{-4}$. Figure 2 shows the average computation times for the various algorithms, taken over 100 trials for each algorithm. In all cases the simulations are performed on a Pentium II PC with a clock speed of 333 MHz.

As evident from Fig. 2, the geometric Newton's method using analytic gradients and Hessians displays the fastest average convergence, followed by the BFGS quasi-Newton method as invoked by fminu( ). Unlike the traditional Newton's method for functions in Euclidean space, quasi-Newton methods employ an approximate Hessian that is much faster to compute than the actual Hessian. The algorithm with the slowest average convergence is the steepest descent method with exact step size. If we instead use the step-size estimate for the line-search procedure, the average convergence time improves dramatically by nearly a factor of 3.

To investigate the effects of programming language and compiler on algorithm performance, Fig. 2 also shows the average convergence times for the steepest descent and Newton's method, in which the algorithms are now implemented in C++. The order of performance of the three algorithms remains unchanged, although in this case there is now only a moderate difference in performance between using the exact and estimated step sizes in steepest descent.

We now consider the effects of scaling the values of N, Q, and W, which were chosen randomly before from a uniform distribution

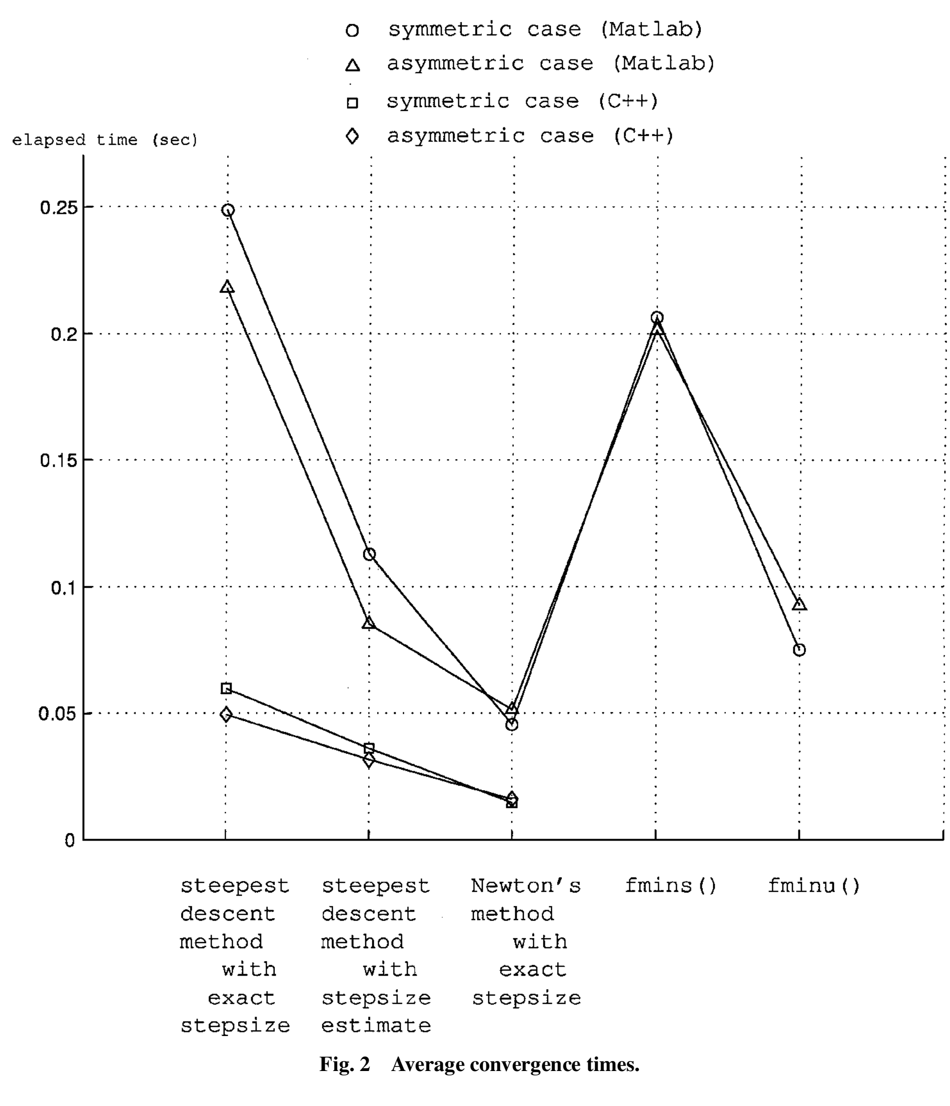

Fig. 2 Average convergence times.

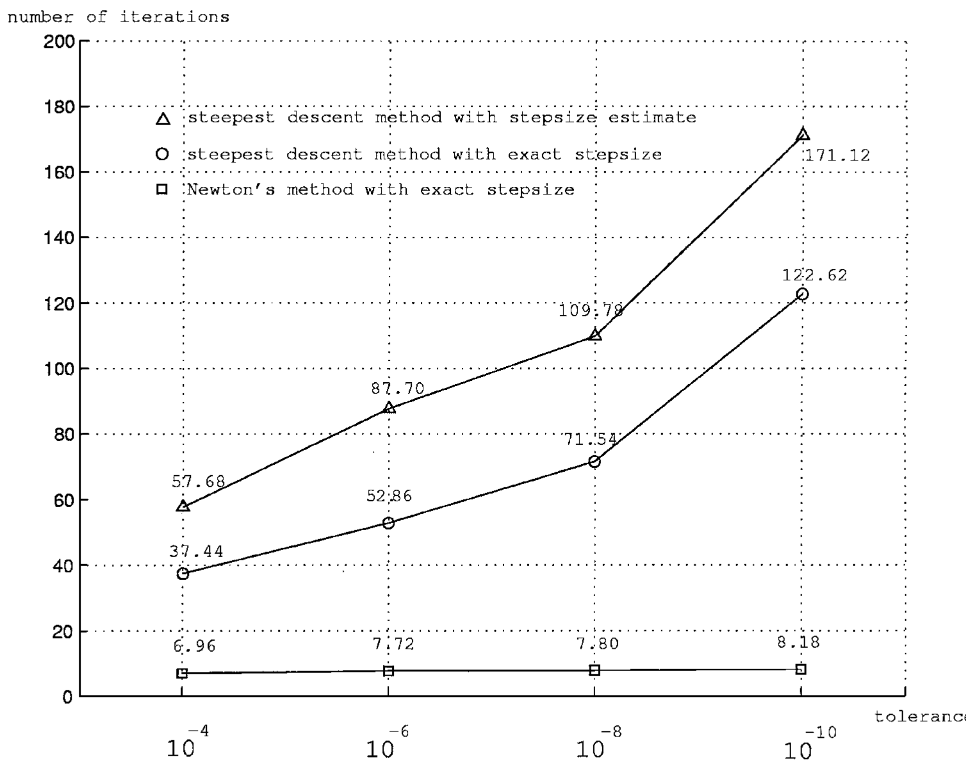

Fig. 3 Number of iterations vs stopping criterion tolerance.

between 0 and 1. Observe that scaling the matrices has the net effect of producing a relative scaling between the first and second terms of the objective function, i.e., the objective function now is of the form

$$J_c(\Theta) = tr(c\Theta N\Theta^T Q - 2\Theta W) \tag{76}$$

where $c > 0$ is an arbitrary constant scale factor. Experimental results indicate that the choice of c has no effect on the average convergence times for the various algorithms. Rather, the tolerance used in the stopping criterion has the greatest effect. Because for all of the algorithms convergence time is proportional to the number of iterations, Fig. 3 graphs the average number of iterations as a function of the stopping tolerance criterion for steepest descent with exact and estimated step sizes and Newton's method.

Figure 2 also compares the average convergence times of steepest descent with exact step size, Newton's method with exact step size, and MATLAB's fmins( ) and fminu( ) for the case of asymmetric N and Q. It also compares the relative performance of the steepest descent method with exact step size and Newton's method with exact step size when the algorithms are implemented in C++. As in the symmetric case, Newton's method has the best average performance, whereas steepest descent with exact step size has the worst average performance.

Finally, we observe that typical convergence times for the C++ implementation of Newton's method result in computational performance of approximately 60 cycles per second. We expect the computation times for these algorithms can be significantly reduced with improved hardware and software.

Conclusions

In this article we have presented a class of numerical optimization algorithms for solving the GPS-based attitude determination problem. By exploiting both the underlying geometric and algebraic structure of the group of rotation matrices $SO(3)$, efficient algorithms that generalize to $SO(3)$ the method of steepest descent and Newton's method were developed. Methods for analytically computing the exact step size in the line-search procedure, as well as for efficiently computing a step-size estimate, were also outlined.

The numerical performance of these algorithms is compared with those of traditional optimization algorithms intended for minimizing objective functions in Euclidean space. As our primary performance benchmarks, the objective function is first formulated in exponential coordinates and minimized using both the Nelder–Mead-type simplex search method and the BFGS quasi-Newton method. Results of numerical experiments suggest that our geometric descent algorithms, in particular Newton's method, offer significant performance advantages over traditional optimization algorithms intended for minimizing objective functions in Euclidean space.

As pointed out in the Introduction, the specific problem we address is admittedly an idealized mathematical one and ignores a number of practical issues that must be resolved either a priori or simultaneously, e.g., cycle ambiguity resolution, the multipath problem, structural distortion caused by flexure and thermal effects, tropospheric effects, and receiver-specific errors, all of which are topics of current study. Clearly there are many additional issues that need to be considered in developing a complete solution to the GPS-based attitude determination problem, but we believe the geometric perspective offers a fundamentally different and promising approach to attitude determination using GPS.

Acknowledgments

This research was supported by the Engineering Research Center for Advanced Control and Instrumentation at Seoul National University. The suggestions of R. Brockett are also gratefully acknowledged.

References

[1] Cohen, C. E., "Attitude Determination," *Global Positioning System: Theory and Applications Vol. II*, edited by B. W. Parkinson and J. J. Spilker Jr., Vol. 164, Progress in Astronautics and Aeronautics, AIAA, Washington, DC, 1996, pp. 519–538.

[2] Wahba, G., "A Least Squares Estimate of Spacecraft Attitude," *SIAM Review*, Vol. 7, No. 3, 1965, p. 409.

[3] Nadas, A., "Least Squares and Maximum Likelihood Estimates of Rigid Motion," IBM Research Rept., RC6945, Yorktown, Hts., NY, 1978.

[4] Brockett, R. W., "Least Squares Matching Problems," *Linear Algebra and Its Applications*, Vol. 122, 1989, pp. 761–777.

[5] Arun, K. S., Huang, T. S., and Blostein, S. D., "Least Squares Fitting of Two 3-D Point Sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 9, No. 5, 1987, pp. 698–700.

[6] Hanson, R. J., and Norris, M. J., "Analysis of Measurements Based on the Singular Value Decomposition," *SIAM Journal of Scientific and Statistical*

Computing, Vol. 2, No. 3, 1981, pp. 363–373.

[7]Markley, F. L., "Attitude Determination Using Vector Observations and the Singular Value Decomposition," *Journal of the Astronautical Sciences*, Vol. 36, No. 3, 1988, pp. 245–158.

[8]Li, Z. X., Gou, J., and Chu, Y., "Geometric Algorithms for Workpiece Localization," *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 6, 1998, pp. 864–878.

[9]Kim, J., "Numerical Optimization on the Rotation Group," M.S. Thesis, School of Mechanical and Aerospace Engineering, Seoul National Univ., Korea, Dec. 1998.

[10]Cohen, C. E., Cobb, H. S., and Parkinson, B. W., "Two Studies of High Performance Attitude Determination Using GPS: Generalizing Wahba's Problem for High Output Rates and Evaluation of Static Accuracy Using a Theodolite," *Proceedings of ICON GPS-92*, Institute of Navigation, Washington, DC, Sept. 1992, pp. 1197–1203.

[11]Wertz, J. R. (ed.), *Spacecraft Attitude Determination and Control*, 1st ed., Vol. 73, Reidel, Dordrecht, The Netherlands, 1973, pp. 427, 428.

[12]Shuster, M. D., "A Survey of Attitude Representations," *Journal of the Astronautical Sciences*, Vol. 41, No. 4, 1993, pp. 439–517.

[13]Bar-Itzhack, I. Y., "REQUEST: A Recursive Quest Algorithm for Sequential Attitude Determination," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 5, 1995, pp. 1034–1038.

[14]Mortari, D., "Energy Approach Algorithm for Attitude Determination from Vector Observations," *Journal of the Astronautical Sciences*, Vol. 45, No. 1, 1997, pp. 41–55.

[15]Mortari, D., "Euler-q Algorithm for Attitude Determination from Vector Observations," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 328–334.

[16]Axelrad, P., and Ward, L. M., "Spacecraft Attitude Estimation Using the Global Positioning System: Methodology and Results for RADCAL," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 6, 1996, pp. 1201–1209.

[17]Axelrad, P., and Behre, C. P., "Attitude Estimation Algorithms for Spinning Satellites Using Global Positioning System Phase Data," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 164–169.

[18]Bar-Itzhack, I. Y., Montgomery, P. Y., and Garrick, J. C., "Algorithms for Attitude Determination Using the Global Positioning System," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 6, 1998, pp. 846–852.

[19]Brockett, R. W., "Dynamical Systems that Sort Lists, Diagonalize Matrices, and Solve Linear Programming Problems," *Linear Algebra and Its Applications*, Vol. 146, 1991, pp. 79–91.

[20]Chu, M. T., "On the Continuous Realization of Iterative Processes," *SIAM Review*, Vol. 30, No. 3, 1988, pp. 375–389.

[21]Bloch, A. M., Brockett, R. W., and Ratiu, T., "A New Formulation of the Generalized Toda Lattice Equations and Their Fixed Point Analysis via the Moment Map," *Bulletin of the American Mathematical Society*, Vol. 23, 1990, pp. 477–485.

[22]Helmke, U., and Moore, J. B., *Optimization and Dynamical Systems*, Springer–Verlag, New York, 1994, Chaps. 1–4.

[23]Brockett, R. W., "Differential Geometry and the Design of Gradient Algorithms," *Proceedings of the Symposium on Pure Mathematics*, Vol. 54, American Mathematical Society, Providence R1, 1993, pp. 69–92.

[24]Curtis, M. L., *Matrix Groups*, Springer–Verlag, New York, 1984, Chaps. 1–5.

[25]Belinfante, J. G., and Kolman, B., *A Survey of Lie Groups and Lie Algebras with Applications and Computational Methods*, Siam, Philadelphia, 1972, Chap. 1.

[26]Boothby, W., *An Introduction to Differentiable Manifolds and Riemannian Geometry*, Academic Press, New York, 1975, Chaps. 3 and 6.

[27]Crassidis, J. L., and Markley, F. L., "New Algorithm for Attitude Determination Using Global Positioning System Signals," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 5, 1997, pp. 891–896.

[28]Oshman, Y., and Markley, F. L., "Minimal Parameter Attitude Matrix Estimation from Vector Observations," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 4, 1998, pp. 595–602.