2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Particle Filtering on the Euclidean Group

Junghyun Kwon, Minseok Choi, Changmook Chun, and F. C. Park

***Abstract*— We address general filtering problems on the Euclidean group $SE(3)$. We first generalize, to stochastic nonlinear systems evolving on $SE(3)$, the particle filter of Liu and West [1] for simultaneously estimating the state and covariance. The filter is constructed in a coordinate-invariant way, and explicitly takes into account the geometry of $SE(3)$ and $P(n)$, the space of symmetric positive definite matrices. An experimental case study involving vision-based robot end-effector pose estimation is also presented.**

I. Introduction

This paper presents a generalization, to stochastic systems evolving on $SE(3)$, of the particle (or sequential Monte Carlo) filter of Liu and West [1] for simultaneously estimating the state and model parameters. Here the geometry of $SE(3)$ and $P(n)$, the space of $n \times n$ symmetric positive definite matrices, plays a key role, and our results are also relevant to generalizations of other particle filtering algorithms. An experimental case study of our filtering framework, involving vision-based pose estimation of a robot manipulator, is also presented.

The practical advantages of particle filtering as an alternative to, *e.g.*, the extended Kalman filter (EKF) for nonlinear systems, have by now been well-documented in the literature (see, *e.g.*, [2]). Aside from a few exceptions that we mention below, however, none of the previous particle filtering works address systems that evolve on curved spaces such as $SE(3)$. In principle one could apply EKF after choosing a set of suitable local coordinates; under assumptions of small noise such methods may work locally. However, globally and under large noise assumptions, their performance cannot be assured. The usual problems associated with lack of coordinate invariance (*e.g.*, both the noise distribution and filter performance depend on the choice of local coordinates, and switching between different coordinate charts must be managed) will moreover persist, making such approaches at best cumbersome and unreliable.

The first paper to explicitly address Monte Carlo filtering on Lie groups is [3], who generalize the sequential importance sampling particle filter to state equations of the form

$$dX = e^V X\, dt, \quad dV = dW, \tag{1}$$

where X is an element of a matrix Lie group, V an element of its corresponding Lie algebra, and W a diffusion process defined on the Lie algebra. The associated measurement equation includes an additive noise term of the form $y = h(X) + \eta$, where η denotes zero-mean Gaussian noise. Case studies involving $SO(3)$ and $SE(3)$ are also presented. Earlier groundbreaking studies by Chirikjian *et al.* on Brownian motions on $SO(3)$ and $SE(3)$ in a robotics context, and stochastic differential equation models of the kinematics of mobile robots and surgical needles, as well as exact solutions to the associated Fokker-Planck equations, convolution formulas on $SE(3)$, and their applications to motion planning and accuracy analysis, have also been presented in [4], [5], [6], [7]. Geometric diffusions on $SO(3)$ also appear in the recent works of Srivastava [8], [9], on object recognition and tracking.

Among the previous works, only [3] and [8] explicitly address particle filtering on $SE(3)$. In this paper we shall consider systems whose state equations are more general than either [3] or [8]. We also construct the filter in a coordinate-invariant way, to avoid the pitfalls associated with local coordinate-based approaches. This involves consideration of a number of geometric issues on $SE(3)$ and $P(n)$, *e.g.*, distance metrics, sample means, discretization and integration of differential equations, and coordinate-invariant constructions of probability distributions on these spaces.

The paper is organized as follows. Section 2 describes the mathematical framework for the general filtering problem on $SE(3)$, and the generalization to $SE(3)$ of the particle filtering algorithm of [1]. Section 3 illustrates the methodology for our experimental case study involving vision-based robot end-effector pose estimation.

II. Particle Filtering on the Euclidean Group

We first consider the following general setting. Let $\mathbf{G}$ be an m-dimensional matrix Lie group and $\mathbf{g}$ its corresponding matrix Lie algebra, with basis elements $E_, \ldots, E_m \in \mathbf{g}$. The state equations and measurements are assumed to be in left-invariant form (the development for right-invariant systems is analogous, and is not repeated):

$$dX = X \cdot A(X)\, dt + X \sum_{i=1}^{m} b_i(X) E_i\, dw_i \tag{2}$$

$$dy = c(X)\, dt + d\eta, \tag{3}$$

where $X \in \mathbf{G}$ is the state, $y \in \Re^p$ is the measurement vector, the maps $A : \mathbf{G} \to \mathbf{g}$, $c : \mathbf{G} \to \Re^p$, and $b_i : \mathbf{G} \to \Re$ are assumed $\mathcal{C}^2$, and $dw_i \in \Re$, $d\eta \in \Re^p$ denote independent Wiener processes. The objective is to estimate the current state $X(t)$ from the measurements y up to the current t.

Financial support for this research was provided in part by the BK21 Program in Mechanical Engineering and the Institute of Advanced Machinery and Design at Seoul National University, and by the Center for Intelligent Robotics at KIST.

J. Kwon, M. Choi, and F. C. Park are with the School of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, Korea jhkwon@robotics.snu.ac.kr, kontiki@robotics.snu.ac.kr, fcp@snu.ac.kr

C. Chun is with the Intelligent Robotics Research Center, Korea Institute of Science and Technology, Seoul 136-791, Korea changmook@kist.re.kr

Vector space particle filtering algorithms have now become firmly entrenched among the tools of probabilistic robotics. Various measure-theoretic issues for performing particle filtering on Lie groups are discussed in [3], together with a discussion of the basic methods of particle filtering. Any geometrically well-defined particle filtering algorithm on a group **G** involves consideration of the following issues: (i) constructing appropriate notions of distributions on **G**; (ii) the metric structure of **G**, and formulas for the sample mean; (iii) discretization and propagation of the state equations on **G**.

A. Distributions and Sample Means

The Special Euclidean Group: The problem of constructing probability distributions on $SE(3)$ is addressed in detail in Wang and Chirikjian [10]. In particular, they show how to construct concentrated local distributions on $SE(3)$ in a coordinate-invariant way, so as to have all the properties typically associated with the Gaussian (*e.g.*, the convolution of Gaussians results in another Gaussian); this is achieved by establishing an equivalence between highly concentrated Gaussian distributions on $SE(3)$ and $se(3)$.

We now discuss formulas for the sample mean on $SE(3)$. There is extensive literature on distance metrics on $SE(3)$ that we do not recount here; we instead cite the main result of Moakher [11], who shows that the mean rotation (in the Euclidean sense) of N rotations $\{R_1, \ldots, R_N\}$, defined as the $R \in SO(3)$ that minimizes $\sum_{n=1}^{N} \|R_n - R\|^2$, where $\|\cdot\|$ denotes the Frobenius norm, is given by the orthogonal projection of $\bar{R} = \sum_{n=1}^{N} \frac{R_n}{N}$. This can be explicitly evaluated as

$$R = \begin{cases} VU^\top, & \text{if } \det(\bar{R}^\top) > 0, \\ VHU^\top, & \text{otherwise,} \end{cases} \tag{4}$$

where U and V are obtained from the singular value decomposition of $\bar{R}^\top$, *i.e.,* $\bar{R}^\top = U\Sigma V^\top$, and $H = \mathrm{diag}[1, 1, -1]$. The sample mean on $SE(3)$ can be obtained by augmenting the previous sample mean formula for $SO(3)$ with the traditional algebraic sample mean for the translational component in $SE(3)$.

The Symmetric Positive-Definite Matrices: Models may contain unknown static parameters (for example, in (2), the covariance S of the noise term, or even the parameters A and b themselves may be unknown), and a common technique to addressing such problems is to include the parameters as part of the state vector, *e.g.*, [1]. In this subsection we focus on the estimation of the unknown covariance S. Covariance matrices are characterized by being symmetric and positive-definite; we denote this space by $P(n)$, and note that it is a subgroup of the general linear group $GL(n)$. The Riemannian structure of $P(n)$ is discussed at length in [12] and [13]; here we present only relevant formulas.

$P(n)$ admits the structure of a metric space in the standard way, *i.e.*, the distance between two points A and B is defined as the infimum of the lengths of piecewise-differentiable curves from A to B, denoted $\delta(A, B)$. The geodesic $\gamma(t) : [0, 1] \to [A, B]$ connecting A and B is given by

$$\gamma(t) = A^{1/2}\big(A^{-1/2}BA^{-1/2}\big)^t A^{1/2} = \Gamma_{A^{1/2}}\big(\Gamma_{A^{-1/2}}(B)^t\big) \tag{5}$$

where Γ_g is defined as $\Gamma_g(p) = gpg^T$, with g is a fixed element in $GL(n)$ and $p \in P(n)$. From this the distance $\delta(A, B)$ between A and B is given by

$$\delta(A, B) = \big(\sum_{i=1}^{n} \log^2 \lambda_i\big)^{1/2}, \tag{6}$$

where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of the matrix AB^{-1}. Since AB^{-1} is similar to $A^{-1/2}(AB^{-1})A^{1/2} = \Gamma_A^{1/2}(B^{-1}) > 0$, the eigenvalues of AB^{-1} are all positive, and hence $\log \lambda_i$ is well defined for each i. Note that $\delta(A, \gamma(t)) = t\delta(A, B)$.

Sample Means and Covariances on $P(n)$: The mean of $p_1, \ldots, p_N \in P(n)$ in the **Euclidean** sense is simply the arithmetic mean $\sum_{i=1}^{N} p_i$. Since $P(n)$ is convex, the arithmetic mean lies in $P(n)$. From a geometric perspective it is pointed out by [12] that the arithmetic mean has a number of disadvantages. The **intrinsic mean** of N given symmetric positive matrices $p_1, \ldots, p_N$ is defined as

$$\arg\min_{\mu \in P(n)} \sum_{i=1}^{N} \delta(\mu, p_i)^2, \tag{7}$$

which is unique for $P(n)$. Given only two points A and B, the intrinsic mean can analytically be derived as $\gamma(\frac{1}{2})$ with $\gamma(t)$ given in (5). For three or more points, the intrinsic mean must be found via optimization. For this purpose the gradient of (7) is given by $\sum_{i=1}^{N} \log(\mu p_i^{-1})$. Details of the steepest descent algorithm are given in [12].

Numerical computation of the intrinsic covariance matrix of a set of points in $P(n)$ is also discussed in [13]. Given N elements, $p_1, \ldots, p_N$, and an intrinsic mean μ, the covariance matrix relative to μ is defined by

$$\Sigma_\mu = \frac{1}{N-1} \sum_{i=1}^{N} v_k v_k^T, \tag{8}$$

where v_k is the tangent vector at μ such that the geodesic $\gamma(t)$ goes through $\gamma(0) = \mu$ and $\gamma(1) = p_k$ with $\dot{\gamma}(0) = v_k$. Based on these mean and covariance formulas, the generalized normal distribution on $P(n)$ is constructed by taking the curvature into account; see [13] for details.

B. Discretization of the State Equations

The main issue in the discretization of the general state equations (2), with initial conditions $X(0) = X_0 \in \mathbf{G}$, where $X \in \mathbf{G}$ and $A(X, t) \in \mathbf{g}$, is that at each time step $X(t)$ remains on the group. Beginning with the pioneering work of Crouch and Grossman [14], considerable literature exists on the subject (see [15] and the references cited therein). The primary motivation in these works is to generalize Runge-Kutta and other numerical integration methods developed for ODEs on $\Re^n$ to general Lie groups. The simplest first-order

discretization, and one that is for the most part sufficient for our purposes, is the exponential Euler discretization given by

$$X_{i+1} = X_i \exp\left(A(X,t)\Delta t + \sum_{j=1}^{m} b_j(X) E_j \sqrt{\Delta t}\epsilon_{i+1,j} \right), \tag{9}$$

where each $\epsilon_i = (\epsilon_{i,1}, \ldots, \epsilon_{i,m})$ is an m-dimensional zero-mean Gaussian with specified covariance matrix S.

C. Particle Filtering with Unknown Covariance

We now generalize the combined state-parameter estimation particle filter of Liu and West [1] to $SE(3)$, where the parameter space is restricted to be $P(n)$ (*i.e.*, the noise covariance S is assumed unknown *a priori*). The observation equation is a generic function on $\mathbf{G} = SE(3)$ and has an additive Gaussian noise:

$$dX = X \cdot A(X)\, dt + X \sum_{i=1}^{m} b_i(X) E_i\, dw_i \tag{10}$$

$$y = h(X) + \eta, \tag{11}$$

where $A(X,t)$ is an element of $\mathbf{g} = se(3)$. η is Gaussian white noise which is independent of dw_i's.

The filtering problem on $SE(3)$ that we consider is the estimation of the state $X(t)$ and S, the covariance of dw_i's, given the time series of measurements, $y(t_0)$, $y(t_1)$, …, $y(t_n)$. One means of simultaneously estimating the pose and the relevant covariance parameters is by embedding the covariance parameters into the state. This is a fairly common technique in system identification, and the mathematical and engineering justification for this technique in the context of particle filtering is covered in, *e.g.*, [1] and the references cited therein.

Since in our case the unknown parameter S evolves on $P(n)$, any kernel smoothing introduced in [1] must be performed to always ensure that S lies on $P(n)$. If there are N parameter particles $\theta^{(1)}, \ldots, \theta^{(N)}$, the mean $\bar{\theta}$ and covariance $\Sigma_{\bar{\theta}}$ can be calculated from (7) and (8). The kernel mean, $a\theta^{(i)} + (1-a)\bar{\theta}$, can be understood as the corresponding point on the geodesic $\gamma(t)$ connecting $\theta^{(i)}$ and $\bar{\theta}$ with $\gamma(0) = \theta^{(i)}$ and $\gamma(1) = \bar{\theta}$; that is, $a\theta^{(i)} + (1-a)\bar{\theta} = \gamma(1-a)$, which can be calculated from (5). The Gaussian sampling with the determined kernel mean and covariance can be realized following the method in [13].

With the above geometric constructions, the particle filtering algorithm for simultaneous covariance and state estimation with kernel smoothing can be described as follows (by a slight abuse of notation, in what follows we continue to use the vector space kernel mean notation for the geometric kernel mean of (5)).

Algorithm:

1) **Initialization:** $t = 0$
   - Set number of particles N.
   - Set δ between $0.95 \sim 0.99$.
   - For $i = 1, \ldots, N$, draw the states $X_0^{(i)}$ and the covariance parameters $\theta_0^{(i)}$ for S from the priors $\rho(X_0)$ and $\rho(\theta_0)$, respectively.
2) **Importance sampling step**
   a) Set $t = t + 1$.
   b) Compute the intrinsic mean $\bar{\theta}_{t-1}$ and covariance $\Sigma_{\bar{\theta}_{t-1}}$ of $\{\theta_{t-1}^{(i)}, i = 1, \ldots, N\}$ from (7) and (8).
   c) For $i = 1, \ldots, N$, draw $X_t^{(*i)} \sim \rho(X_t | X_{t-1}^{(i)}, \theta_t^{(*i)})$, *i.e.,*
      i) Determine the kernel mean $a\theta_{t-1}^{(i)} + (1-a)\bar{\theta}_{t-1}$ from (5).
      ii) Draw $\theta_t^{(*i)}$ from $N(a\theta_{t-1}^{(i)} + (1-a)\bar{\theta}_{t-1}, h^2\Sigma_{\bar{\theta}_{t-1}})$ where $a = (3\delta - 1)/2\delta$ and $h^2 = 1 - a^2$.
      iii) Generate the Gaussian ϵ_t from $N(0, \theta_t^{(*i)}\Delta t)$, and propagate $X_{t-1}^{(i)}$ to $X_t^{(*i)}$ via (9).
   d) For $i = 1, \ldots, N$, weight each draw by
   $$w_t^{(i)} \propto \rho(y_t | X_t^{(*i)}), \tag{12}$$
   e) For $i = 1, \ldots, N$, normalize the importance weights:
   $$\tilde{w}_t^{(i)} = w_t^{(i)} \left[\sum_{j=1}^{N} w_t^{(j)} \right]^{-1} \tag{13}$$
3) **Selection step (resampling)**
   a) Resample from $\{X_t^{(*1)}, \ldots, X_t^{(*N)}\}$ and $\{\theta_t^{(*1)}, \ldots, \theta_t^{(*N)}\}$ with probability proportional to $\tilde{w}_t^{(i)}$ to produce a random sample $\{X_t^{(1)}, \ldots, X_t^{(N)}\}$ and $\{\theta_t^{(1)}, \ldots, \theta_t^{(N)}\}$.
   b) For $i = 1, \ldots, N$, set $w_t^{(i)} = \tilde{w}_t^{(i)} = \frac{1}{N}$.
4) **Go to the importance sampling step**

For practical implementation purposes, in our case study we will assume S is block-diagonal of the form $S = \text{diag}\{S_1, S_2\}$, where $S_1 \in P(3)$ and $S_2 \in P(3)$ are covariances for the orientation and position, respectively. In this case two covariance parameters need to be introduced; the extension of the above algorithm to this case is straightforward.

The output of the algorithm is a set of samples that can be used to approximate the posterior distribution of $X(t)$ on the Lie group:

$$\pi(X_t | y_t) \approx \hat{\pi}(X_t | y_t) = \frac{1}{N} \sum_{i=1}^{N} \delta(X_t - X_t^{(i)}), \tag{14}$$

where $\delta(\cdot)$ denotes the Dirac delta function. To obtain the optimal estimate $\hat{X}(t)$, we use the sample mean formula on $SE(3)$ as given in (4).

III. Application to Vision-Based End-Effector Pose Estimation

As an experimental case study, we consider the problem of estimating the end-effector pose of a robot using vision sensors. For robots in typical laboratory or industrial settings, one usually has available accurate kinematic models of the robot (often obtained via a kinematic calibration procedure involving the use of laser, ball-bar, or other position measurement sensors of reasonably high accuracy), together with

accurate joint encoders, so that it is enough to evaluate the forward kinematics for reliably determining the end-effector pose as:

$$X = e^{A_1 q_1} e^{A_2 q_2} \cdots e^{A_6 q_6} M \tag{15}$$

where $X \in SE(3)$ represents the location of the end-effector frame relative to the fixed frame, $q_1, \ldots, q_6$ denote the joint variables, $A_1, \ldots, A_6 \in se(3)$ denote the kinematic parameters, and $M \in SE(3)$ denotes the end-effector location when the robot is in its home position.

Things change considerably for low-cost personal robots operating in unstructured home settings. The kinematic models provided by the manufacturer are often inaccurate and unreliable, and high precision sensors are usually not available for performing kinematic calibration. Further compounding matters is that the robots themselves are subject to greater manufacturing tolerances (and thus greater errors), resulting in a greater degree of parametric errors, backlash, friction, joint and link elastic deformations, and tracking control errors. Torque-level control is also typically not feasible, making attempts to construct accurate dynamic models and associated observers moot.

The vision-based pose estimation algorithm presented here is intended for a low-cost personal robot. We also assume a low-cost noisy vision sensor (such as a webcam) capable of measuring a pre-specified set of feature points on the end-effector, and a nominal kinematic model with potentially large parameter errors. Because of the relatively noisy vision sensor, accurate end-effector pose estimation based on vision measurements alone is inadequate (not least of all because the classical depth ambiguity problem cannot be easily resolved in our setting); it is essential to make use of the vision measurements in conjunction with the available kinematic model.

For the vision sensor, we assume that a single perspective projection camera is available, and that the image plane coordinates can be measured for a fixed number n of feature points on the end-effector frame. Let $Y = (y_1, \ldots, y_n)$ denote the measurements from the vision sensor, where each y_i represents the 2-D homogeneous coordinates of feature point i in the image plane and can be represented as a function of X, *i.e.*, $y_i = K \left[\begin{array}{cc} R & -R\bar{c} \end{array} \right] X x_i$, where $R \in SO(3)$ is the camera rotation matrix, $\bar{c} \in \Re^3$ is the camera center in the fixed frame, x_i represents the 3-D homogeneous coordinates of each feature point attached to the end-effector with respect to the end-effector frame, and $K \in \Re^{3\times 3}$ is the camera calibration matrix determined by the focal length f and the principal point (p_x, p_y) on the image plane. The measurement equation can be expressed in the form $Y = h(X)$, where $h(\cdot)$ represents the aforementioned perspective projection mapping onto the 2-D image plane.

We choose to express the kinematic state equations in right-invariant form, *i.e.*,

$$\begin{aligned} dX &= V(u_1, \ldots, u_6) \cdot X dt + dW \cdot X & (16)\\ V(u_1, \ldots, u_6) &= A_1 u_1 + Ad_{e^{A_1 q_1}}(A_2) u_2 + \ldots & (17)\\ Y &= h(X) + \eta. & (18) \end{aligned}$$

(a) (b)

Fig. 1. (a) The Cycloid 3 currently being used in experimental studies. (b) The "3Com HomeConnect PC Digital WebCam", "Krypton GCS300" stereo camera, and its probe.

where Ad denotes the adjoint mapping on $SE(3)$ (*i.e.*, $Ad_X(V) = XVX^{-1}$ for $X \in SE(3)$ and $V \in se(3)$), and $u_i = \dot{q}_i$, $i = 1, \ldots, 6$ the joint velocity inputs. The joint velocities are taken to be the control inputs. Equations (16) and (18) are discretized as

$$\begin{aligned} X_t &= \exp(V(u_1, \ldots, u_6)\Delta t + dW) \cdot X_{t-1} & (19)\\ Y_t &= h(X_t) + \eta. & (20) \end{aligned}$$

Experiments are performed with the Cycloid 3 shown in Fig. 1(a), a small, inexpensive 19-DOF humanoid developed for educational and entertainment purposes. For the experiment we fix all joints except for the one waist joint and the three joints in the right arm, so that we can effectively regard the Cycloid as a 4-DOF open chain. We use the 3Com HomeConnect PC Digital WebCam shown in Fig. 1(b), with 640×480 resolution, as the vision sensor.

To evaluate the experimental performance of our pose estimation algorithm, we use the Krypton GCS300 real-time stereo camera and sensor probe system (see Fig. 1(b)) to measure the actual 3-D pose of the end-effector frame with respect to the fixed frame. The probe, now regarded as the end-effector frame, is attached to the right hand of the Cycloid, and the camera frame is taken to be the fixed frame. The four infrared LED's on the probe are used as feature points for the end-effector frame.

We use the direct linear algorithm of [16] to obtain the camera matrix of the webcam, from a large set of point correspondences between the measured 3-D points of the four feature points and their 2-D corresponding points on the image plane. The variance of the additive Gaussian noise η is heuristically set to 4. Nominal values for the Cycliod's highly imprecise kinematic parameters (which are not provided by the manufacturer) are obtained via direct measurement, and are shown in Table I.

TABLE I

KINEMATIC PARAMETERS FOR EACH JOINT OF THE CYCLOID 3.

Joint	ω	v
Joint 1	(0.9651,-0.1075,-0.2389)	(-199.3,-1834.4,20.2)
Joint 2	(-0.1700,-0.9507,-0.2593)	(-1790.9,311.7,31.7)
Joint 3	(0.1993,-0.2909,0.9358)	(-646.7,-344.4,30.7)
Joint 4	(-0.1700,-0.9507,-0.2593)	(-1793.6,331.5,-39.1)

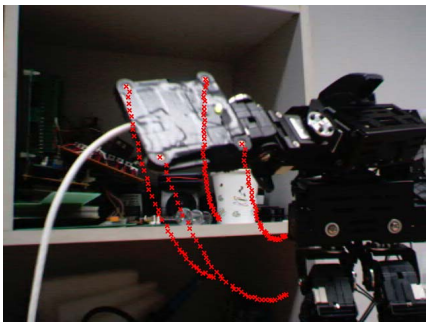

Fig. 2. Trajectories of the feature points of the end effector frame on the webcam image plane.

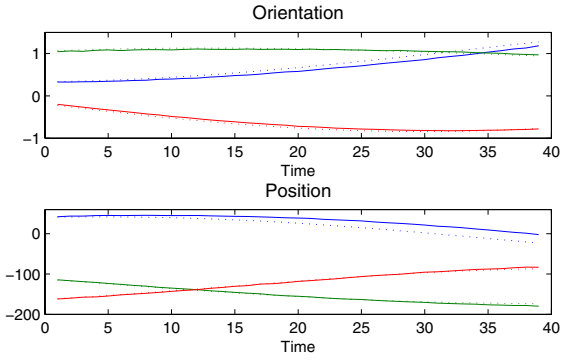

Fig. 3. The measured (solid line) and nominal (dotted line) end-effector frames. Top: Blue, green, and red lines represent the three independent elements of the logarithm of the rotation matrix. Bottom: Blue, green, and red lines represent (1,4), (2,4) and (3,4) elements of the frame.

The Cycloid is manipulated for 20 seconds according to a set of pre-specified joint control inputs, while the webcam captures the end-effector feature points every 0.5 seconds. Trajectories of the feature points captured by the webcam, *i.e.*, $Y_{1:40}$, are shown in Fig. 2. Fig. 3 shows the end-effector pose trajectory $X_{1:40}$ as measured by the Krypton GCS300. The end-effector pose trajectory calculated from the nominal forward kinematics equation (15), $\tilde{X}_{1:40}$, is also shown together as the dotted line, clearly showing large errors from the actual end-effector pose. As for the orientation components of the end-effector frame, three independent elements of the logarithm of the rotation matrix are shown.

5000 particles are used in the filtering algorithm. Each particle $X_0^{(i)}$ is initialized from the nominal end-effector pose $\tilde{X}_0$ as $X_0^{(i)} = \tilde{X}_0 \cdot \exp(N)$ with some specific covariance matrix whose values are rather small for N which represents the zero mean Gaussian noise on $se(3)$. We set δ to 0.99 and initialize the covariance matrices. For S_1, a symmetric matrix is uniformly sampled between 0 and 0.00001 for the diagonal elements and between -0.000005 and 0.000005 for the off-diagonal elements. Then, we perform the singular value decomposition for the sampled symmetric matrix, into $U\Sigma V^T$, and replace V with U to make the sampled symmetric matrix positive definite. For S_2, the uniform distribution between 0 and 1 is used for the diagonal elements and between -0.5 and 0.5 for the off-diagonal elements.

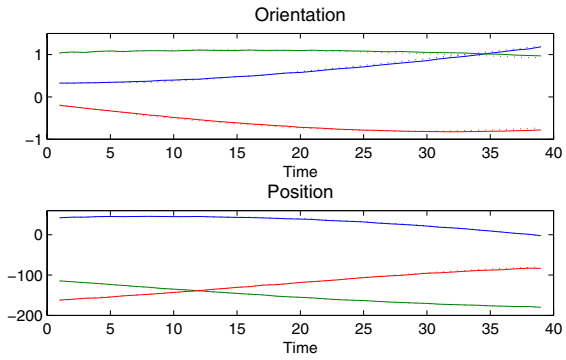

Fig. 4. The measured (solid line) and estimated (dotted line) end-effector frames. Top: Blue, green, and red lines represent the three independent elements of the logarithm of the rotation matrix. Bottom: Blue, green, and red lines represent (1,4), (2,4) and (3,4) elements of the frame.

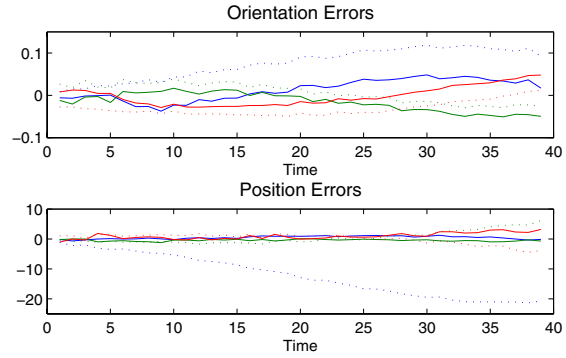

Fig. 5. The estimation errors (solid line) and the errors between the measured and nominal end-effector frames (dotted line). Top: Blue, green, and red lines represent the errors for the three independent elements of the logarithm of the rotation matrix. Bottom: Blue, green, and red lines represent the errors for (1,4), (2,4) and (3,4) elements of the frame.

The estimated end-effector pose trajectory $\hat{X}_{1:40}$ is shown in Fig. 4, while Fig. 5 shows the estimation errors. The errors between the measured end-effector frame and that obtained from the nominal forward kinematics frame are shown together as the dotted line for comparison. The norms of the errors for $\hat{X}_{1:40}$ are 0.2495 for the orientation component and 9.6908 for the position, while those for $\tilde{X}_{1:40}$ are 0.5454 and 88.1265, respectively. The orientation component of the errors are clearly less severe compared to position errors. Both orientation and position estimation accuracy are improved, with the position components showing particularly large improvement. Finally, Fig. 6 shows the elements of the estimated covariances.

IV. Conclusions

This paper has investigated particle filtering on the Euclidean group $SE(3)$. Assuming the covariance of the state space noise model is *a priori* unknown, we generalize the particle filter of Liu and West [1] for simultaneously estimating state and model parameters to $SE(3) \times P(n)$, where $P(n)$ denotes the space of $n \times n$ symmetric positive-definite matrices. Both the theoretical underpinnings of the filter and the implementation and performance issues are

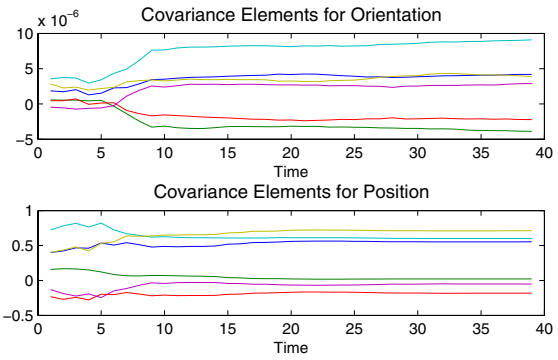

Fig. 6. The estimated covariance parameters.

discussed in the context of an experimental study on vision-based end-effector pose estimation. It is important to account for the geometry of $SE(3)$ and $P(n)$ in any generalization of vector space particle filtering algorithms. In this regard we discuss the metric structure, formulas for the sample mean, the construction of distributions, discretization and propagation of the state equations, and other geometric issues involving the two groups.

The framework and experimental case study presented in the paper represent just one aspect of the larger problem of Monte Carlo estimation on Lie groups, and as such several open problems and extensions can be posed in a more general setting, *e.g.*, extending the estimation algebra approach of [17] to stochastic systems on Lie groups, determining the minimum variance linear filter for bilinear systems on Lie groups (see [18] for the corresponding minimum variance linear filter for bilinear systems on vector spaces), generalization of other particle filtering algorithms, and improving the computational efficiency of the algorithms.

Finally, an extension to other applications, and other Lie groups, would also be of interest. In particular, the problem of attitude estimation using global positioning system (GPS) data as measurements can be kinematically formulated (see [19]) as a set of state equations on $SO(3)$ of the form $dR = RA(t)\,dt + R\,dW$, where $R \in SO(3)$, $dW \in so(3)$, and $A(t) \in so(3)$ is allowed to be time-varying. The noise accounts for errors due to various factors such as parameter uncertainty, unmodelled dynamics, etc. The measurement equations can be conveniently expressed in matrix form $Y = S^\top RB + V$, where the measurements are arranged as elements of the matrix $Y \in \mathbb{R}^{n\times m}$, $B = (b_1, b_2, \ldots, b_m) \in \mathbb{R}^{3\times m}$ and $S = (\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_n) \in \mathbb{R}^{3\times n}$ are given matrices obtained from the measurement data, and the elements of $V \in \mathbb{R}^{n\times m}$ represent standard independent identically distributed Gaussian noise.

Another application involves the simultaneous estimation of the needle position and noise covariance in needle steering [5]. The uncertainties in the needle tip motion can be captured by a stochastic kinematic state equation of the form $dX = XAdt + XdW$, where $dW \in se(3)$ is standard Wiener noise with a covariance S, and $A = (\omega, v) \in se(3)$ is of the form $\omega = (\kappa u_1, 0, u_2)$, $v = (0, 0, u_1)$; here κ is the curvature that describes the amount of bending, and u_1 and u_2 are respectively the translational and rotational insertion speeds along the needle at the point of insertion. The measurement equations for the needle tip position $y \in \Re^3$, obtained via electromagnetic tracking, are corrupted by noise of the form $y = p + n$, where p denotes the tip position and n represents three-dimensional Gaussian noise with covariance Q.

Other matrix Lie groups that arise in various engineering applications include the group $SL(3)$ of volume preserving linear transformations, which plays a prominent role in vision and medical imaging, as does the affine group.

REFERENCES

[1] J. Liu and M. West, "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. Freitas, and N. Gordon, Eds., pp. 97-223, New York: Springer-Verlag, 2001.

[2] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filter for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 174-188, 2002.

[3] A. Chiuso and S. Soatto, "Monte Carlo filtering on Lie groups," in *Proc. 39th IEEE Conf. Decision and Control*, vol. 1, pp. 304-309, 2000.

[4] G. S. Chirikjian and A. B. Kyatkin, *Engineering Applications of Noncommutative Harmonic Analysis*. Boca Raton: CRC Press, 2000.

[5] W. Park, J. S. Kim, Y. Zhou, N. J. Cowan, A. M. Okamura, and G. S. Chirikjian, "Diffusion-based motion planning for a nonholonomic flexible needle model," *Proc. IEEE Int. Conf. Robotics Autom.*, pp. 4611-4616, 2005.

[6] Y. Zhou and G. S. Chirikjian, "Probabilistic models of dead-reckoning error in nonholonomic mobile robots," *Proc. IEEE Int. Conf. Robotics Autom.*, pp. 1594–1599, 2003.

[7] Y. Zhou and G. S. Chirikjian, "Nonholonomic motion planning using diffusion of workspace density functions," *Proc. Int. Symp. Advances in Robot Dynamics and Control*, Nov. 16-21, Washington, D.C., 2003.

[8] A. Srivastava, "Bayesian filtering for tracking pose and location of rigid targets," *Proc. SPIE Aerosense*, Orlando, FL, April 2000.

[9] A. Srivastava and E. Klassen, "Monte Carlo extrinsic estimators of manifold-valued parameters," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 299-308, 2002.

[10] Y. Wang and G. S. Chirikjian, "Error propagation on the Euclidean group with application to manipulator kinematics," *IEEE Trans. Robotics*, vol. 22, no. 4, pp. 591-602, 2006.

[11] M. Moakher, "Means and averaging in the group of rotations," *SIAM J. Matrix Analysis & Applications*, vol. 24, no. 1, pp. 1-16, 2002.

[12] P. Fletcher and S. Joshi, "Principal Geodesic Analysis on Symmetric Spaces: Statistics of Diffusion Tensors," *Proc. ECCV'04 Workshop Computer Vision Approaches to Medical Image Analysis*, pp. 87-98, 2004.

[13] C. Lenglet, M. Rousson, R. Deriche and O. Faugeras, "Statistics on the Manifold of Multivariate Normal Distributions: Theory and Application to Diffusion Tensor MRI Processing," *J. Math. Imaging Vis.*, vol. 25, no. 3, pp. 423-444, 2006.

[14] P. Crouch and R. Grossman, "Numerical integration of ordinary differential equations on manifolds," *J. Nonlinear Sci.*, vol. 3, pp. 1-33, 1993.

[15] A. Iserles and H. Munthe-Kaas, "Lie group methods," in *Acta Numerica (2000)*, Cambridge: Cambridge University Press, 2000.

[16] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge: Cambridge University Press, 2000.

[17] S. S. -T. Yau and Y. T. Lai, "Explicit solution of DMZ equation in nonlinera filtering via solution of ODEs," *IEEE Trans. Automatic Control*, vol. 48, no. 3, pp. 505-508, 2003.

[18] L.C. Vallot, "Filtering for bilinear systems," M.S. Thesis, MIT, Cambridge, MA, 1981.

[19] F. C. Park, J. Kim, and C. Kee, "Geometric descent algorithms for GPS-based attitude determination," *AIAA J. Guidance, Control, and Dynamics*, vol. 23, no. 1, pp. 26-33, 2000.