Sampling-based Algorithm for Filtering using Markov chain Approximations

Pratik Chaudhari Sertac Karaman Emilio Frazzoli

Abstract— **This paper proposes a filtering algorithm applicable to a large class of continuous-time, continuous-state stochastic dynamical systems. The proposed approach is largely inspired by recent advances in asymptotically-optimal sampling-based motion planning algorithms such as PRM* and RRT*. It is based on an incremental construction of a sequence of Markov chain approximations constructed in such a way that they satisfy certain local consistency conditions. We prove that the trajectories of these Markov chains converge in distribution to the trajectories of the original stochastic system and that the optimal filter calculated on these Markov chains converges to the optimal nonlinear continuous time filter. Simulations verify the convergence of the discrete approximations to the original stochastic system. Finally, a number of examples are provided that show the performance of the filtering algorithm.**

I. Introduction

Stochastic filtering, initially developed in the works of Wiener and Kolmogorov, is an essential problem in systems and control theory. Among the main results in this area is the Kalman filter, which has dominated signal processing and optimal state estimation of linear systems for decades now. Non-linear filtering methods like Extended Kalman filter, and the Unscented Kalman filter [1] have been introduced to expand its applicability. This family of algorithms is however limited to systems with Gaussian additive noise and Gaussian conditional densities.

In many real problems, nonlinearities and non-Gaussian noise prevents us from getting closed form expressions for the optimal filter. The seminal paper by Gordon, Salmond and Smith [2] introduced the bootstrap filter which forms the basis for a class of general filters known as sequential Monte-Carlo methods [3]. They utilize a large number of random samples (called as particles) to represent arbitrary posterior distributions and are propagated in time using importance sampling techniques. The crucial aspect of particle filtering is estimating a good posterior to sample from. Techniques like resampling posterior [4] to reduce variance of particles and adaptive sampling [5] result in the algorithm being flexible and applicable to a wide class of nonlinear and non-Gaussian models. On the other hand, it is necessary to tune the filtering algorithm to the given problem for robust performance [6].

Continuous time filtering algorithms have also received wide attention in literature, starting from the Kalman-Bucy filter for continuous time linear systems with Gaussian additive noise. More recent results on continuous time particle filters are inspired by weak approximations of solutions of stochastic differential equations (SDEs) and come with explicit rates of convergence [7]. On a related note, the solution of a SDE can also be approximated by a set of ordinary differential equations to perform filtering as shown in [8]. Elsewhere, branching and interacting particle systems in continuous time have also been applied to the nonlinear filtering problem [9]. Numerical solutions to the partial differential equations arising from the Zakai equation and the Kushner-Stratonovich equation have been used to perform continuous time nonlinear filtering [10]. These applications of these algorithms are however limited due to computational intractability or non-recursive nature.

The primary motivation of this paper was to take steps towards creating a general filtering algorithm but eliminate the need to tune it to different systems. In particular, we focus on the continuous time nonlinear filtering problem in our formulation. Our results draw inspiration from two main areas. Firstly, we are interested in the Markov chain approximation approach, developed by Kushner [11], that has been proposed as a method to generate discrete approximations of ordinary SDEs with continuous representations of states, controls, and observations. The method is, in principle, applicable to a wide class of problems in optimal stochastic control [11] and optimal estimation [12], but its applications in the literature are scarce, possibly due to the computational complexity of *a priori* discretization of continuous spaces.

In a different context, the curse of dimensionality has been shown to be inevitable in robot motion planning problems, i.e., the problem of finding a dynamically feasible trajectory around obstacles so as to reach a goal region is PSPACE-hard [13]. Yet, algorithms with probabilistic guarantees such as Probabilistic RoadMaps (PRM) [14] or the Rapidly-exploring Random Tree (RRT) [15] have been shown to work quite effectively in returning a feasible solution in high-dimensional configuration spaces by effectively discretizing the said space based on random sampling. In [16], two novel motion planning algorithms were proposed under the name of PRM* and RRT*, which are shown to be both computationally efficient and asymptotically optimal. In particular, the RRT* algorithm has been successfully applied to many challenging motion planning problems involving high-dimensional configuration spaces [17], complex dynamical systems [18]. Similar sampling techniques were also used to generate MDPs for stochastic optimal control [19].

We leverage these recent results in asymptotically optimal motion planning algorithms and apply them to construct incrementally refined Markov chain approximations of stochastic systems. The key idea behind this construction is

The authors are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA. Email: {pratikac,sertac,frazzoli}@mit.edu

Preprint submitted to 51st IEEE Conference on Decision and Control .
Received March 7, 2012.

to discretize time and state space at a rate similar to that of PRM* and the RRT* algorithms, and get the transition probabilities of the chain using local consistency ideas of the Markov chain approximation method. We prove that the trajectories of these successive approximations converge in distribution to the trajectories of the original stochastic dynamical system. We propose an algorithm to solve the nonlinear optimal filtering problem using these approximations and also provide an example of *maximum a posteriori* trajectory estimation. The resulting algorithms draw their features from both motion planning and Markov chain approximation method and are, (i) fairly general, i.e., designed for a large class of stochastic dynamical systems, (ii) easy to implement even for complex dynamical systems and (iii) do not need to be explicitly tuned for different problems or for platforms with different computational capabilities due to their incremental nature.

This paper is organized as follows. The continuous-time nonlinear filtering problem for a stochastic dynamical system is defined in Section II. Section III is devoted to some preliminary background and results. The construction of Markov chain approximations is presented in Section IV. Section V gives details of the application of these Markov chains for filtering. Convergence proofs for the proposed algorithms are outlined in Section VI. Experimental comparisons are reported and discussed in Section VII. We provide a novel direction for future work in Section VIII.

II. Problem Definition

Let $\mathbb{R}$ denote the set of real numbers and $\mathbb{R}^{n\times k}$ denote the set of all $n \times k$ real valued matrices. Consider a stochastic differential equation of the form

$$dx(t) = f(x(t))\,dt + F(x(t))\,dw(t), \quad x(0) = x_0, \qquad (1)$$

where (i) $x(t) \in \mathbb{R}^d$ for all $t \geq 0$, (ii) $f : \mathbb{R}^d \to \mathbb{R}^d$, $F : \mathbb{R}^d \to \mathbb{R}^{d\times k}$ are Borel-measurable functions, (iii) the stochastic process $\{w(t) : t \geq 0\}$ is the standard k-dimensional Brownian motion, and the random variable x_0 is bounded with probability one. A solution to the differential form presented in Equation (1) is a stochastic process $\{x(t) : t \geq 0\}$ that constitutes a solution to the following integral equation:

$$x(t) = x_0 + \int_0^t f(x(\tau))d\tau + \int_0^t F(x(\tau))\ dw(\tau),\ \forall\ t \geq 0,$$

where the second term on the right hand side is the usual Itô integral [20]. We tacitly assume throughout the paper that the functions $f(\cdot)$ and $F(\cdot)$ are bounded and continuous functions to guarantee weak existence and weak uniqueness for the solutions of Equation (1).

In the standard nonlinear filtering problem [11], [20] one attempts to estimate the process $\{x(t); t \geq 0\}$ using data available till time t, defined by $\mathcal{Y}_t := \{y(s) : s \leq t\}$, where $\{y(t) : t \geq 0\}$ is a solution to the stochastic differential equation of the form

$$dy(t) = g(x(t))\,dt + G(x(t))dv(t), \qquad (2)$$

where $g : \mathbb{R}^d \to \mathbb{R}^m$ and $G : \mathbb{R}^d \to \mathbb{R}^{m\times l}$ are Borel-measurable functions, and $\{v(t) : t \geq 0\}$ is a l-dimensional Brownian motion independent of the stochastic process $\{x(t) : t \geq 0\}$. Similarly, we assume that the functions $g(\cdot)$ and $G(\cdot)$ are bounded and continuous to guarantee weak existence and weak uniqueness of solutions to Equation (2). As in [11], we formulate the problem such that the system evolves inside a compact subset, denoted by $\mathcal{S}$, of $\mathbb{R}^d$. The process is stopped if it hits the boundary of $\mathcal{S}$. That is, define

$$\tau := \inf\{s : x(s) \notin \mathcal{S}^o\},$$

where, $\mathcal{S}^o$ denotes the interior of $\mathcal{S}$. Then, the definition of the problem is given as follows.

Problem 1 *Given a set $\mathcal{Y}_t := \{y(s) : s \leq t\}$ of observations generated by process* (2)*, find an estimate $\hat{x}(t)$ such that (i) $\mathbb{E}[\|x(t) - \hat{x}(t)\|^2]$ is minimized and (ii) the random variable $\hat{x}(t)$ is square integrable and H_t-measurable, where H_t is the σ-algebra generated by $\mathcal{Y}_t$.*

It is well known that the error-minimizing state estimate $\hat{x}(t)$ based on observations $\mathcal{Y}_t$ is,

$$\hat{x}(t) = \mathbb{E}[x(t) \,|\, \mathcal{Y}_t].$$

In fact, this equation forms the basis of the Fujisaki-Kallianpur-Kunita equation of filtering theory [20]. Let us also note that the solution of the filtering problem can be given by a representation as follows [21]. If $\tilde{x}(t)$ is a process with the same distribution as that of $x(t)$ but independent of $(x(t), y(t))$, and $\phi(\cdot)$ is any continuous real-valued function, then the solution of the filtering problem is,

$$\mathbb{E}[\phi(x(t)) \,|\, \mathcal{Y}_t] = \frac{\mathbb{E}[R(t)\phi(\tilde{x}(t)) \,|\, \mathcal{Y}_t]}{\mathbb{E}[R(t) \,|\, \mathcal{Y}_t]}, \qquad (3)$$

where

$$R(t) = \exp\left[\int_0^t g(\tilde{x}(s))^T dy(s) - \frac{1}{2}\int_0^t |g(\tilde{x}(s))|^2 ds\right].$$

Essentially, in our approach, the Markov chain approximation method generates a process $\tilde{x}(t)$ which has the same law as the original process $x(t)$. Equation (3) which is really the limiting version of Bayes' rule then gives the optimal filter.

III. Preliminaries

A. Discrete Markov Chains

A *Markov chain* is denoted by the tuple $\mathcal{M} = (S, P)$, where $S \subset \mathcal{S}$ is a finite set of states and $P(\cdot \,|\, \cdot) : S \times S \to \mathbb{R}_{\geq 0}$ is a function that denotes the transition probabilities, i.e., the function $P(z \,|\, z')$ is the probability that the next state is z given that the current state is z'. As a conditional probability mass function, P satisfies $\sum_{z\in S} P(z \,|\, z') = 1$ for all $z' \in S$. The trajectory of a Markov chain starting from an initial state z_0 is denoted by the sequence $\{\xi_i : i \in \mathbb{N}\}$ of random variables that satisfy (i) $\xi_0 = z_0$ and (i) $\mathbb{P}(\xi_{i+1} = z \,|\, \xi_i = z') = \mathbb{P}(z \,|\, z')$ for all $z, z' \in S$ and all $n \in \mathbb{N}$.

B. Optimal Filter on a Markov chain

The optimal filtering problem for a discrete Markov chain is very similar to Problem 1. Given a Markov chain $\mathcal{M} =$

Preprint submitted to 51st IEEE Conference on Decision and Control .
Received March 7, 2012.

(S, P) and a set $Y_k = \{y_i : i = 1, 2, \ldots, k\}$ of discrete-time observations coming from an equation of the form $y_k = g(\xi_k) + G(\xi_k)\tilde{v}$, where, $\tilde{v}$ is unit-variance white Gaussian noise, we can calculate the conditional distribution $\phi_k(z) = \mathbb{P}(\xi_k = z \,|\, Y_k)$ to be

$$\phi_n(z) = \sum_{z' \in S} \mathbb{P}(\xi_n = z, \xi_{n-1} = z' \,|\, Y_n),$$

which can be written using recursive Bayes' rule as

$$\phi_n(z) = \eta \textstyle\sum_{z' \in S} \mathbb{P}(y_n \,|\, \xi_n = z, \xi_{n-1} = z') \times \mathbb{P}(\xi_n = z \,|\, \xi_{n-1} = z') \; \phi_{n-1}(z'), \quad (4)$$

where η is a normalization constant and $\phi_0(z)$ is the initial distribution of states. Note that the probability $\mathbb{P}(y_n \,|\, \xi_n = z, \xi_{n-1} = z')$ becomes $\mathbb{P}(y_n \,|\, \xi_n = z)$ under our observation equation. This formulation is similar to estimation on Hidden Markov Models [22] except for the fact that observations come from an observation space instead of a finite set. So long as we can calculate $\mathbb{P}(y_n \,|\, \xi_n = z)$, the same formulae hold. Also note that the observations in Equation (2) are often approximated [12] by discretizing them at times $k\delta$ as

$$y_k = g(x(k\delta))\delta + G(x(k\delta))[v(k\delta) - v(k\delta - \delta)],$$

which is the conventional discrete observation model

$$y_k = \tilde{g}(x_k) + G(x_k)\; \tilde{v}_k,$$

with $\tilde{g}(x_k) = g(x(k\delta))\; \delta$ and $\tilde{v}_k = v(k\delta) - v(k\delta - \delta)$ being the white Gaussian noise constructed from the continuous time Brownian motion $v(t)$. Note that $\tilde{v}_k$ need not be Gaussian noise in the above equation. $v(t)$ however needs to be Brownian motion to formally guarantee existence and uniqueness of solution to Equation (2).

C. The Markov Chain Approximation Method

Let $\mathcal{M} = (S, P, T)$ be a Markov chain where T is the set of functions $\Delta t : S \to \mathbb{R}_{>0}$ that associate a time interval to each state in S. The function Δt is called the function of interpolating times, or a *holding time* for short. Roughly, $\Delta t(z)$ is the time that the chain spends at state z, before making another transition. Given an initial state $z_0 \in S$, let $\{\xi_i; i \in \mathbb{N}\}$ denote the trajectory of the the Markov chain $\mathcal{M}$ starting from z_0. $\xi(\cdot)$ is the *continuous-time interpolation* of such trajectories under holding times Δt i.e.,

$$\xi(\tau) = \xi_i \quad \text{for all } \tau \in [t_i, t_{i+1}),$$

where $t_i = \sum_{j=1}^{i} \Delta t(\xi_j)$. For any realization $\xi(t, \omega)$ of the stochastic process $\{\xi(t); t \in \mathbb{R}_{\geq 0}\}$, the function $\xi(\cdot, \omega)$ is continuous from the left and has limits from the right. i.e., $\xi(t, \omega) \in D^d[0, \infty)$. Hence, ξ can be thought of as a random mapping that takes values in the function space $D^d[0, \infty)$.

Given a continuous time interpolation, we can come up with conditions under which the trajectories of a sequence of Markov chains converge to the trajectories of the original process described by Equation (1). Let $\{\mathcal{M}_n; n \in \mathbb{N}\}$, where $\mathcal{M}_n = (S_n, P_n, T_n)$, denote a sequence of Markov chains. For each $n \in \mathbb{N}$, let $\{\xi_i^n; i \in \mathbb{N}\}$ be the trajectory of $\mathcal{M}_n$ with initial state distributed according to some distribution π_n. The sequence of Markov chains $\{\mathcal{M}_n; n \in \mathbb{N}\}$ is said to be *locally consistent* with the original system described by Equation (1) if the following criteria are satisfied for all $z \in S$.

$$\circ \quad \lim_{n \to \infty} \Delta t_n(z) = 0, \quad (5)$$

$$\circ \quad \lim_{n \to \infty} \frac{\mathbb{E}[\xi_{i+1}^n - \xi_i^n \,|\, \xi_i^n = z]}{\Delta t_n(z)} = f(z), \quad (6)$$

$$\circ \quad \lim_{n \to \infty} \frac{\mathrm{Cov}[\xi_{i+1}^n - \xi_i^n \,|\, \xi_i^n = z]}{\Delta t_n(z)} = F(z)F(z)^T. \quad (7)$$

As stated in the following theorem, under mild technical assumptions, local consistency implies the convergence of continuous-time interpolations of the trajectories of the Markov chain to the trajectories of the stochastic dynamical system described by Equation (1).

Theorem 2 (Theorem 10.4.1 in [11]) *Assume that $f(\cdot)$ and $F(\cdot)$ are bounded and continuous. Let $\{\mathcal{M}_n; n \in \mathbb{N}\}$ be a sequence of Markov chains that are locally consistent with stochastic dynamical system described by Equation* (1)*. For each $n \in \mathbb{N}$, let $\{\xi_n(t); t \in \mathbb{R}_{\geq 0}\}$ denote the continuous-time interpolation to the trajectory of $\mathcal{M}_n$. Then, $(\xi_n(\cdot))$ has a subsequence that converges in distribution to $(x(\cdot))$ that satisfies*

$$x(t) = x_0 + \int_0^t f(x(s))ds + \int_0^t F(x(s))\; dw(s),$$

where x_0 is distributed according to $\lim_{n \to \infty} \pi_n$, π_n being the prior distribution of the initial state on $\mathcal{M}_n$.

Example Consider the system $dx = -x\; dt + \sigma\; dw$ on some bounded interval of $\mathcal{S} \subset \mathbb{R}_{\geq 0}$ with a regular discretization of distance h. In this example, any state x in the Markov chain is only connected with its neighbors $x - h$ and $x + h$. Roughly, the Markov chain transitions to the right using only diffusion whereas it uses both drift and diffusion to go left.

$$P(x + h \,|\, x) = \frac{\sigma^2/2}{c} \quad \text{and} \quad P(x - h \,|\, x) = \frac{\sigma^2/2 + hx}{c}$$

These transition probabilities sum up to 1, i.e., $c = (\sigma^2 + hx)$. Finally, local consistency conditions are satisfied if we choose $\Delta t_h = \; c^{-1}h^2$.

IV. Construction of Approximating Chains

This section describes some primitive procedures and the algorithm for the construction of discrete Markov chains.

A. Primitive procedures

a) Sampling: The `Sample` procedure returns states sampled independently and uniformly from $\mathcal{S} \subset \mathbb{R}^d$.

b) Neighboring states: Given $z \in \mathcal{S}$ and a finite set $S \subseteq \mathcal{S}$ of states, the procedure $\texttt{Near}(z, S)$ returns the set of all states that are within a distance of $r = \gamma\; (\log n/n)^{1/d}$ from z, i.e.,

$$\texttt{Near}(z, S) = \left\{ z' \in S,\; z' \neq z : \|z' - z\| \leq \gamma \left(\frac{\log n}{n}\right)^{1/d} \right\}$$

where $n = |S|$, $d = \dim(S)$ and $\gamma > 0$ is a constant that will be specified in Section VI. Given a state $z \in S$, let Z_{near} be the set of states returned by $\texttt{Near}(z, S)$.

Preprint submitted to 51st IEEE Conference on Decision and Control .
Received March 7, 2012.

c) Time Intervals: Given a state $z \in S$, the procedure $\texttt{ComputeHoldingTime}(z, S)$ returns a holding time

$$\Delta t(z) = \frac{r^2}{\|F(z)F^T(z)\|_2 + r\|f(z)\|_2},$$

where r is as given in the procedure $\texttt{Near}(z, S)$. The expression of $\Delta t(z)$ is motivated by $\Delta t = \frac{\texttt{distance}}{\texttt{average velocity}}$ and ensures that the system approximately remains within a ball of radius r centered at z after traveling for time $\Delta t(z)$. As seen from the local consistency conditions, we only need $\Delta t(z) \to 0$ which happens as $r \to 0$ i.e., $n \to \infty$ making the exact expression is quite flexible.

d) Transition Probabilities: Local consistency conditions translate into a linear program for finding the transition probabilities. However, we can also use a local Gaussian to get the probabilities as follows. Given a state $z \in \mathcal{S}$ and a finite set $Z_{\text{near}} \subset \mathcal{S}$, the $\texttt{ComputeTransProb}(z, Z_{\text{near}}, \Delta t(z))$ returns a function $p(\cdot \,|\, z)$ which is computed as follows. Let $\mathcal{N}_{\mu,\Sigma}(\cdot)$ denote the density of the (possibly multivariate) Gaussian distribution with mean μ and variance Σ. Define the transition probabilities as $p(z' \,|\, z) = \eta\, \mathcal{N}_{\mu,\Sigma}(z')$ where $\mu = z + f(z)\Delta t(z)$ and $\Sigma = F(z)F(z)^T \Delta t(z)$ and the constant η ensures $\sum_{z' \in Z_{\text{near}}} p(z' \,|\, z) = 1$. Lemma 3 in Section VI proves that this satisfies local consistency conditions in the limit.

B. Incremental construction of the Markov chain

Algorithm 1 uses the procedures described above to generate the Markov chain. In particular, once it has a chain $\mathcal{M}_n$, it adds the $(n+1)^{\text{th}}$ sample to create a more refined chain $\mathcal{M}_{n+1}$. Algorithm 1 thus creates the sequences of Markov chains upon which we perform filtering.

Algorithm 1: Incremental Markov chain Construction

1 $n \leftarrow 0$;
2 **while** $n < N$ **do**
3 $\quad z \leftarrow \texttt{Sample}()$;
4 $\quad S_n \leftarrow S_{n-1} \cup \{z\}$;
5 $\quad (S_n, P_n, T_n) \leftarrow \texttt{ConnectState}(z, (S_n, P_{n-1}, T_{n-1}))$;
6 $\quad Z_{\text{near}} \leftarrow \texttt{Near}(z, S_n)$;
7 $\quad$ **for** $z_{\text{near}} \in Z_{\text{near}}$ **do**
8 $\quad\quad (S_n, P_n, T_n) \leftarrow \texttt{ConnectState}(z_{\text{near}}, (S_n, P_n, T_n))$;
9 $\quad n \leftarrow n + 1$;
10 **return** (S_N, P_N, T_N);

Algorithm 2: $\texttt{ConnectState}(z, (S, P, T))$

1 $\Delta t(z) \leftarrow \texttt{ComputeHoldingTime}(z, S)$;
2 $Z_{\text{near}} \leftarrow \texttt{Near}(z, S)$;
3 $P(\cdot \,|\, z) \leftarrow \texttt{ComputeTransProb}(z, Z_{\text{near}}, \Delta t(z))$;
4 $T(z) \leftarrow \Delta t(z)$;
5 **return** (S, P, T);

C. Batch construction of the Markov chain

If N samples are drawn before-hand instead of sampling incrementally, we can get rid off lines 6-8 in Algorithm 1. The holding times $\Delta t(z)$ are then a function of the final N. Let us call this version the "batch construction" of the Markov chain. We will use it as an intermediate step in the proof for the incremental algorithm.

Algorithm 3: Batch Markov chain construction

1 $n \leftarrow 0$;
2 **while** $n < N$ **do**
3 $\quad z \leftarrow \texttt{Sample}()$;
4 $\quad S \leftarrow S \cup \{z\}$;
5 $\quad n \leftarrow n + 1$;
6 **for** $z \in S_N$ **do**
7 $\quad (S_N, P_N, T_N) \leftarrow \texttt{ConnectState}(z, (S_N, P_N, T_N))$;
8 **return** (S_N, P_N, T_N);

D. Computational complexity

The `Near` procedure takes worst case $\mathcal{O}(\log n)$ time using approximate nearest neighbor algorithms [23]. Note that the expected number of samples in a ball of radius $\gamma \left(\frac{\log n}{n}\right)^{1/d}$ is $\mathcal{O}(\log n)$. Hence, the `ComputeTransProb` procedure takes $\mathcal{O}(\log n)$ time. The complexity of `ConnectState` is thus $\mathcal{O}(\log n)$ and it executes for an expected $\mathcal{O}(\log n)$ samples in lines 7-8. Thus, the computational complexity of creating the incremental Markov chain in Algorithm 1 is $\mathcal{O}(n\,(\log n)^2)$. This reduces to $\mathcal{O}(n \log n)$ if we sample the states beforehand as in Algorithm 3.

V. Filtering on Markov chain approximations

We use Equation (4) to propagate estimates on the Markov chain constructed in Section IV. This requires that we know the single step transition probabilities, i.e., roughly, the holding times of all states need to be the same. This section details the construction of such a Markov chain.

We use the Markov chain obtained in Section IV with holding times $\Delta t(z)$, possibly different for different $z \in \mathcal{S}$ along with a discretization δ of the time axis to obtain a modified chain $\mathcal{M}_n^\delta = (S_n, P_n, \delta)$ so that the holding time of every state is δ. This corresponds to augmenting the state-space with a time dimension. Given a state $z \in S_n$ and a finite set $Z_{\text{near}} \subset S_n$, the `ComputeTransProb` procedure is replaced by a new $\texttt{ComputeTransProbTime}(z, Z_{\text{near}}, \Delta t, \delta)$ that returns a probability density function over $T_\delta \times Z_{\text{near}}$, where $T_\delta = \{0, \delta, 2\delta, \dots, \}$. This probability density is denoted by $p^\delta(\cdot \,|\, k\delta, z)$ defined for $z \in S$ and $k \in \{1, 2, \dots\}$. Let $p(\cdot \,|\, z) = \texttt{ComputeTransProb}(z, Z_{\text{near}}, \Delta t(z))$ be computed as described in Section IV-A. Then, $p^\delta(\cdot \,|\, k\delta, z)$ is constructed from $p(\cdot \,|\, z)$ as,

$$\begin{aligned} 1 - p^\delta\big(\, k\delta + \delta, z \,|\, k\delta, z\,\big) &= \frac{\delta}{\Delta t(z)} \\ \frac{p^\delta\big(\, k\delta, z \,|\, k\delta, z'\,\big)}{1 - p^\delta\big(\, k\delta + \delta, z \,|\, k\delta, z\,\big)} &= p(z \,|\, z') \end{aligned} \tag{8}$$

A short calculation verifies that Equations (8) also satisfy the local consistency conditions and the modified Markov chain can be used for filtering using Equation (4).

Preprint submitted to 51st IEEE Conference on Decision and Control .
Received March 7, 2012.

Equation (8) suggests that $\delta \leq \min_{z \in S_n} \Delta t(z)$. If the Markov chain is obtained from Algorithm 3, we fix a $\delta = \min_{z \in S_n} \Delta t(z)$ and modify the transition probabilities of every state $z \in S_n$ using Equations (8). On the other hand, if the Markov chain is being constructed incrementally using Algorithm 1, we cannot fix such a δ because $\Delta t(z)$ is decreasing as $n \to \infty$. Instead, we incrementally reduce the time discretization as $\delta_{\text{new}} = \delta_{\text{current}}/2$ and recalculate probabilities for all states in S_n every time we add a new state z_{n+1} that has $\Delta t(z_{n+1}) \leq \delta_{\text{current}}$. Since $\delta \sim \Delta t(z) = \mathcal{O}((\frac{\log n}{n})^{2/d})$ from Section IV-A, the two successive values of n, n_1, and n_2 when we have to recalculate the probabilities are exponentially increasing i.e., $n_2 \sim n_1 2^{d/2}$ which gives an amortized complexity of $\mathcal{O}(n(\log n)^2)$.

Let us note a few features of the modified Markov chain. For a time-homogeneous SDE, the transition probabilities are dependent only on the state and not on any particular time. Hence, it is not necessary to compute the transition probabilities for each time in T_δ separately in the implementation. The filtered estimate calculated using Equation 4 converges to the optimal continuous-time filtered estimate as $\delta \to 0$ and $n \to \infty$. We prove this in Section VI. Also, if observations are discrete with an interval Δ, choose δ to be a factor of Δ along with the constraint that $\delta \leq \min_{z \in S_n} \Delta t(z)$ while getting $\mathcal{M}_n^\delta$ from $\mathcal{M}_n$ and propagate Equation (4) without the observation probability Δ/δ times before incorporating a new observation.

VI. Analysis

The theorems in this section operate along with Theorem 2 to prove that the approximation generated by Algorithm 1 converges, in some suitable sense to the original process described by Equation (1). In particular, they prove that the sequence of Markov chains $\{\mathcal{M}_n; n \in \mathbb{N}\}$ can be generated incrementally using uniform random sampling.

Lemma 3 *The Gaussian approximation presented in the* `ComputeTransProb` *procedure satisfies the local consistency conditions given in equations* (6) *and* (7)*.*

Proof: Let $\phi(x(t))$ denote the probability density without observations of the stochastic dynamics given in Equation (1). The Fokker-Plank equation can then be used to compute this density as,

$$\frac{\partial}{\partial t}\phi(x(t)) = \left[-\frac{\partial}{\partial x}f(x) + \frac{1}{2}\,\frac{\partial}{\partial x}F(x)F^T(x)\right]\phi(x(t))$$

The solution for small times Δt can then be written to obtain [24],

$$P(x', t+\Delta t \,|\, x, t) = \frac{1}{\sqrt{2\pi F(x)F^T(x)\Delta t}}\ \exp\left(-\frac{1}{2}\frac{[x'-x-f(x,t)\Delta t]^2}{F(x)F^T(x)\Delta t}\right)$$

The Gaussian for the transition probabilities in the `ComputeTransProb` procedure is thus the small time solution of the Fokker-Planck equation. Also, it can be proved that the number of samples in the neighborhood of every sample (in every grid cell $G_n(i)$ of Theorem 4 to be precise) is increasing [16], i.e. the small time solution of the Fokker-Plank equation becomes exact as $n \to \infty$. ■

Theorem 4 *The Markov chain (S_n, P_n, T_n) returned by Algorithm 3 is locally consistent with the stochastic dynamical system described by Equation 1, with probability one.*

Proof: The transition probabilities are consistent by Lemma 3. Hence, we only need to prove that using Algorithm 3, every state has non-zero probability of transition to another state, i.e. every state is connected to at least one other state or that the Markov chain is irreducible. The analysis here is similar to the analysis in [16], [19].

For each $n \in \mathbb{N}$, divide the state space $\mathcal{S}$ into grid cells with side length $\frac{\gamma}{2}(\log n/n)^{1/d}$ as follows. Define the grid cell $G_n(i)$ for $i \in \mathbb{Z}^d$ as

$$i\left(\frac{\gamma}{2}\frac{\log n}{n}\right)^{1/d} + \left[-\frac{1}{4}\gamma\left(\frac{\log n}{n}\right)^{1/d},\ \frac{1}{4}\gamma\left(\frac{\log n}{n}\right)^{1/d}\right]^d,$$

where $[-a, a]^d$ denotes the d-dimensional cube with side length $2\,a$ centered at the origin. Hence, the expression above translates the d-dimensional cube with side length $\frac{\gamma}{2}(\log n/n)^{1/d}$ to the point with coordinates $i\,\frac{\gamma}{2}(\log n/n)^{1/d}$. Let K_n denote the indices of set of all cells that lie completely inside the state space $\mathcal{S}$, i.e., $K_n = \{i \in \mathbb{Z}^d : G_n(i) \subseteq \mathcal{S}\}$.

We claim that for all large n, all grid cells in K_n contain at least one vertex of S_n. Given an event A, let A^c denote its complement. Let $A_{n,k}$ denote the event that the cell $G_n(k)$ contains a vertex from S_n. Then, for all $k \in K_n$,

$$\begin{aligned}\mathbb{P}\left(A_{n,k}^c\right) &= \left(1 - \frac{(\frac{\gamma}{2})^{-d}}{\mu(\mathcal{S})}\,\frac{\log n}{n}\right)^n \\ &\leq \exp\left(-((\frac{\gamma}{2})^d/\mu(\mathcal{S}))\,\log n\right) = n^{-(\frac{\gamma}{2})^d/\mu(\mathcal{S})},\end{aligned}$$

where $\mu(\mathcal{S})$ denotes Lebesgue measure assigned to $\mathcal{S}$. Let A_n denote the event that all cells $G_n(i)$ contain at least one vertex of S_n. Then,

$$\begin{aligned}\mathbb{P}(A_n^c) &= \mathbb{P}\left(\left(\bigcap\nolimits_{k\in K_n} A_{n,k}\right)^c\right) = \mathbb{P}\left(\bigcup\nolimits_{k\in K_n} A_{n,k}^c\right) \\ &\leq \sum\nolimits_{k\in K_n} \mathbb{P}\left(A_{n,k}^c\right) = |K_n|\, n^{-(\frac{\gamma}{2})^d/\mu(\mathcal{S})},\end{aligned}$$

where the first inequality follows from the union bound and $|K_n|$ denotes the cardinality of the set K_n. Merely calculating the maximum number of cubes that can fit into $\mathcal{S}$, the latter quantity can be bounded by

$$|K_n| \leq \frac{\mu(\mathcal{S})}{(\frac{\gamma}{2})^d\,\frac{\log n}{n}} = \frac{\mu(\mathcal{S})}{(\frac{\gamma}{2})^d}\,\frac{n}{\log n}.$$

Hence,

$$\begin{aligned}\mathbb{P}\left(A_n^c\right) &\leq \frac{\mu(\mathcal{S})}{(\frac{\gamma}{2})^d}\,\frac{n}{\log n}\,n^{-(\frac{\gamma}{2})^d/\mu(\mathcal{S})} \\ &\leq \frac{\mu(\mathcal{S})}{(\frac{\gamma}{2})^d}\,n^{1-(\frac{\gamma}{2})^d/\mu(\mathcal{S})},\end{aligned}$$

which is summable for all $\gamma > 2\ (2\,\mu(\mathcal{S}))^{1/d}$. Hence, by the Borel-Cantelli lemma, the probability that A_n^c occurs infinitely often is zero, which implies that the probability

that A_n occurs for all large n is one. Since the radius of the ball in the procedure `Near` is $\gamma(\frac{\log n}{n})^{1/d}$, every state z is connected to at least one other state. Finally, since, $\Delta t(z) \to 0$ as $n \to \infty$ we have proved that Algorithm 3 is locally consistent. ■

Theorem 5 *Incremental construction of the approximating chain using Algorithm 1 is also locally consistent for large n, with probability one.*

Proof: The proof of connectivity of the Markov chain is the same as the proof of Theorem 4 whereas Equations (6)-(7) are satisfied for any state $z \in \cup_{i\in\mathbb{N}} S_i$ by Lemma 3. We only to show that Equation (5) is satisfied, i.e, $\Delta t(z)$ for any state z that is added to the Markov chain at some iteration, say i, goes to zero. Note that, calling `ConnectState` on an existing state always results in reduction of $\Delta t(z)$ because n is increasing. We thus essentially prove that z is reconnected to its neighbors (which are changing with n) infinitely often.

Fix some iteration i and some state $z \in S_i$. Let A_n, defined for all $n > i$, denote the event that the state z belongs to $\texttt{Near}(z_n, S_n)$ of the newly node z_n at iteration n. It is thus inside the ball of volume $\gamma^d(\frac{\log n}{n})$ centered at z_n. Hence, $\mathbb{P}(A_n) = \frac{\gamma^d}{\mu(\mathcal{S})}(\frac{\log n}{n})$. Since $\sum \mathbb{P}(A_n) = \infty$ and the event A_n is independent from A_i for all $i \neq n$, Borel-Cantelli lemma implies that $\mathbb{P}(\limsup_{n\to\infty} A_n) = 1$. Hence, any state z is reconnected infinitely often, with probability 1. ■

Theorems 2, 4, and 5 imply that the trajectories of the successive Markov chains (S_n, P_n, T_n) converge in distribution to the trajectories of the system described by Equation (1).

Theorem 6 (see Theorem 4.1 in [12]) *For any continuous real-valued function $\phi(\cdot)$, for any $T < \infty$, if $\xi_n(\cdot)$ is a sequence which converges in distribution to the solution to Equation (1) i.e., $x(t)$, as $n \to \infty$ and is independent of $(x(t), y(t))$, then,*

$$\lim_{n\to\infty} \sup_{t\leq T} \left| \mathbb{E}[\phi(\xi_n(t)) \,|\, \mathcal{Y}_t] - \mathbb{E}[\phi(x(t)) \,|\, \mathcal{Y}_t] \right| = 0.$$

The above theorem coupled with the formula given in Equation (3) proves that the filtered density calculated on the Markov chain $\mathcal{M}_n^\delta$ converges to the optimal nonlinear filtering density as $n \to \infty$ and $\delta \to 0$.

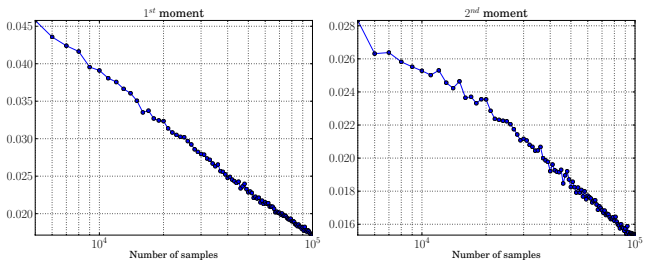
VII. Experiments

This section is devoted to experiments using the algorithms proposed in this paper.

A. Convergence of trajectories

We can numerically verify the results of Theorems 4 and 5 by a Monte-Carlo simulation. Since the distributions of trajectories converge, the distribution of states at any fixed time t also converges. Also, by definition, the moments of the distributions of states at any time t converge, which we will verify. Consider a 2-dimensional single integrator with drift but no observations,

$$\begin{aligned} dx_1 &= -\frac{1}{2}x_1\ dt + 0.03\ dw_1 \\ dx_2 &= -x_2\ dt + 0.03\ dw_2 \end{aligned} \tag{9}$$



(a) (b)

Fig. 2: Figure (a) shows $|\mathbb{E}[\xi_n(T)] - \mathbb{E}[x(T)]|$ versus the number of samples n while Figure (b) shows a similar plot for the 2^{nd} moment, i.e. $|\mathbb{E}[\xi_n(T)^2] - \mathbb{E}[x(T)^2]|$

Figure 1 simulates $50,000$ trajectories of the Markov chain and the actual system dynamics until a time $T = 2$ secs and looks at the distribution of states at five specific time instants. We can also compare the moments of the distribution of the states $x(T)$ and $x_{\text{Markov}}(T)$ for different number of states in the Markov chain. Figure 2 shows the convergence of the error in moments calculated over $50,000$ trajectories with increasing number of states in the Markov chain ranging from $1,000$ to $100,000$.

B. Filtering

We compare the proposed filter with other filtering algorithms like EKF and particle filter on a number of examples in this section.

Consider a drifting ship [25] confined to move within a disc of radius 9 units. A large force $f_i(x(t))$ acts on the ship to make it move inwards if it is moving outwards when it goes out of this disc as shown in Equation (10). The ship is like a 2-dimensional double integrator with forces $f_1(x)$, $f_2(x)$ with observations being range and heading as given in Equation (11). Figures 3 shows that the tracking error is similar to that of the particle filter.

$$f_i(x(t)) = \frac{-50x_i}{\sqrt{x_1^2 + x_2^2}}\ \mathcal{I}_{\{\sqrt{x_1^2+x_2^2}\geq 9\}}\mathcal{I}_{\{x_1x_3+x_2x_4\geq 0\}}$$

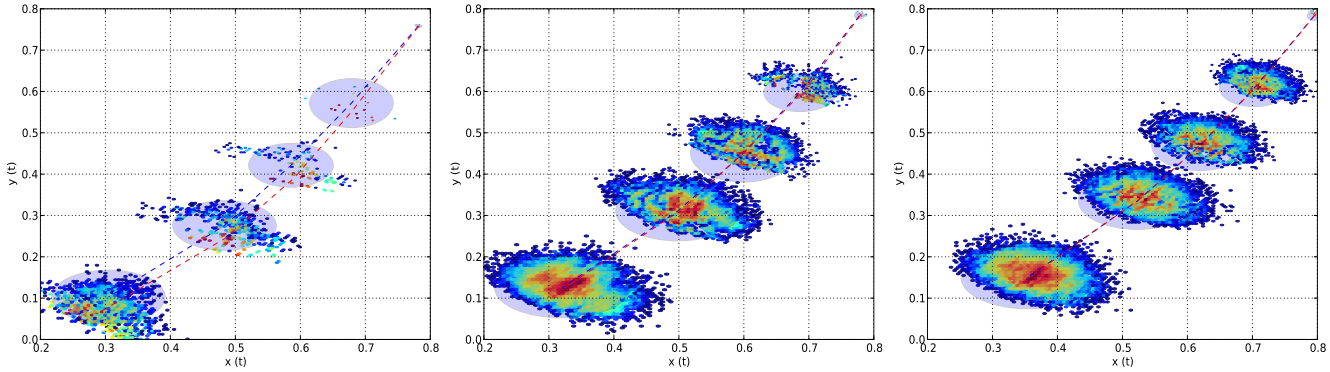
$$\begin{aligned} dx_1 &= x_3\ dt + e\ dw_1 \\ dx_2 &= x_4\ dt + e\ dw_2 \\ dx_3 &= f_1(x)\ dt + e\ dw_3 \\ dx_4 &= f_2(x)\ dt + e\ dw_4 \end{aligned} \tag{10}$$

$$\begin{aligned} dy_1 &= [x_1^2 + x_2^2]^{1/2}dt + e_1\ dv_1 \\ dy_2 &= \tan^{-1}(x_2/x_1)dt + e_2\ dv_2 \end{aligned} \tag{11}$$

Next, we consider a noisy Van der Pol oscillator given by Equation (12). This system is highly non-linear with a stable limit cycle for $\mu > 0$.

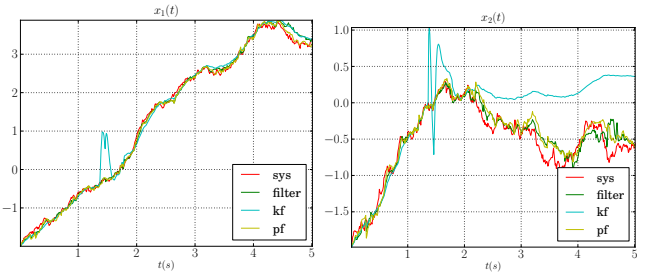
$$\begin{aligned} dx_1 &= x_2\ dt + e_1\ dw_1 \\ dx_2 &= [-x_1 + \mu\ x_1\ (1 - {x_1}^2)]\ dt + e_2\ dw_2 \\ dy &= x_1\ dt + e_3\ dv \end{aligned} \tag{12}$$

The last equation is the scalar observation equation and $\mu = 2$. Figure 4 shows the performance of the sampling filter on this system. Note that this system is typically hard for the EKF which accumulates linearization error due to



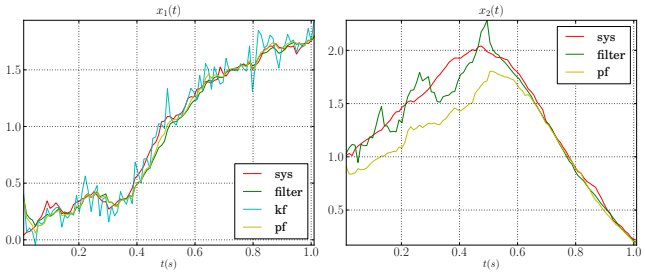
(a) 1000 samples (b) $10,000$ samples (c) $40,000$ samples

Fig. 1: Scatter plots show the distribution of states (x_1, x_2) of the Markov chain at five specific time instants $t \in \{0, 0.3, 0.5, 1.0, 2.0\}$ secs. Translucent ellipses are 3σ ellipses from the simulation of the stochastic system as given in Equation (9). The dotted blue and red lines show the means of the actual and Markov trajectories respectively for $t \in [0, 2]$ secs. The mean trajectories converge i.e. the first moment of the distribution converges as more samples are added. The variance shown as a scatter plot also converges. Both the Markov chain and the original system are started from the nearest state to $(0.8, 0.8)$ in S_n.



(a) $x_1(t)$ (b) $x_2(t)$

Fig. 3: Filter estimate for the drifting ship in Equation (10) with $e = 0.3$, $e_1 = 0.03$ and $e_2 = 0.03$. The EKF error is very large near $(0, 0)$, when the observation nonlinearity is large. The average estimated state error i.e., $\mathbb{E}[\frac{1}{T}\int_0^T \|x - \hat{x}\| dt]$ is 5.02×10^{-3} for the HMM filter, 5.2×10^{-3} for the particle filter both with 100 particles (see Section VII-C) and 1.36×10^{-2} for the EKF.

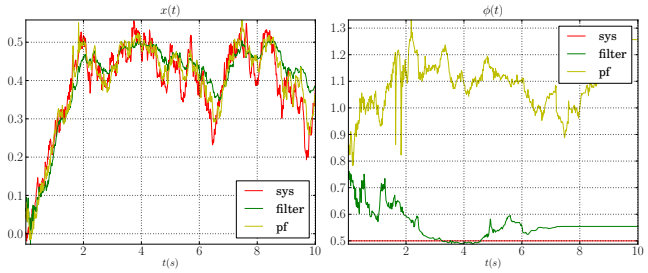


(a) (b)

Fig. 4: $x_1(t)$ and $x_2(t)$ for a Van der Pol oscillator. Mean error of the estimate averaged over 100 runs was 0.1816 for the proposed filter with 100 particles and 0.1849 for the particle filter with 100 particles. The EKF estimate for $x_2(t)$ completely diverges.

varying time scales and, predictably, the EKF estimate of $x_2(t)$ completely diverges. The proposed filter took 0.2 secs to execute while the PF took 0.013 secs for 100 samples with similar average error. This example shows that the proposed filter performs as well as other filters both in terms of estimate and is also computationally tractable.

Next we compare these filters on a modified version of a parameter estimation problem from [26] as given in



(a) (b)

Fig. 5: $x(t)$ and $\phi(t)$ for the parameter estimation problem with $\sigma_x = 0.1$, $\sigma_v = 0.1$ and $\sigma_\phi = 0.1$. Average state error over 100 Monte-Carlo runs was 1.44 for the proposed filter whereas it was 1.878 for the particle filter with 100 samples for both.

Equation (13). The parameter we are estimating is $\phi = 0.5$.

$$\begin{aligned} dx &= x\cos(2\pi\phi x)\ dt + \sigma_x\ dw_1 \\ d\phi &= 0\ dt + \sigma_\phi\ dw_2 \\ dy &= x\ dt + \sigma_v\ dv. \end{aligned} \tag{13}$$

To begin with, it is only known that $\phi \sim \mathcal{N}(0.8, 1)$. Append the state-space with ϕ and inject small noise dw_2 into its dynamics for estimation. This is a hard problem for a particle filter because the conditional density of ϕ given data is not in the exponential family [26] which makes resampling difficult. Figure 5 shows an example run with the particle filter using multinomial sampling. The proposed filter consistently ends up with lower estimation error.

C. Implementation details

Heuristics specific to the filtering problem applied to our Markov chain construction can vastly improve the computational complexity in practice. As proposed in Algorithms 1 and 3, we sample the bounded state-space uniformly. Roughly, this results in the convergence rate depending upon the size of state-space. In order to avoid this, we can concentrate the samples around the estimated posterior to create the Markov chain. In the examples given in Section VII-B, the mean and variance of the prior (assumed to be Gaussian)

Preprint submitted to 51st IEEE Conference on Decision and Control .
Received March 7, 2012.

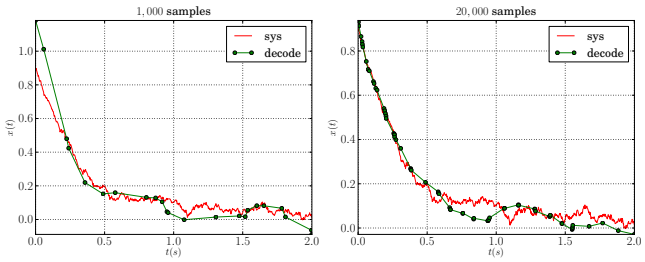
are propagated for a time δ to get the posterior which is used to concentrate samples. We now show an application to the MAP decoding problem where we can directly use the incremental Markov chain without this heuristic.

D. Maximum a posteriori (MAP) trajectory

We will look at discrete time MAP trajectory estimation (decoding) in this section. Given observations till a time t denoted as $Y_t = \{y_1, y_2, \ldots, y_t\}$, it finds the most probable trajectory $\hat{x}_t = \{x_1, \ldots, x_t\}$ i.e.,

$$\hat{x}_t = \arg\max_{\xi} \mathbb{P}(\xi_t \mid Y_t)$$

For a Markov chain $\mathcal{M}_n^\delta = (S_n, P_n, \delta)$, the Viterbi algorithm [27] gives the most probable trajectory after t observations. The incremental Markov chain for filtering constructed in Section V can be used directly in the Viterbi algorithm. The observation probability in this case is given as, $\mathbb{P}(y_k \mid z)$ for $z \in S_n$ which can be obtained from the observation model. Figure 6 shows the decoded trajectory for $x_1(t)$ of a 2D linear system with dynamics $dx_1 = -x_1 dt + \sigma dw_1$ and observations $dy_1 = x_1 dt + \gamma dv_1$. The corresponding equations of $x_2(t)$ are similar.



(a) (b)

Fig. 6: Decoded trajectories with $\sigma = 0.1$ and $\gamma = 0.1$. Total error between the actual and estimated trajectory, calculated as, $\int_0^T \|x(t) - \hat{x}(t)\|_2^2 \, dt$ is 11.34×10^{-3} with $1,000$ samples in Figure (a) whereas it is 3.3×10^{-3} with $20,000$ samples in Figure (b).

VIII. Conclusion

We proposed the Markov chain approximation method as a way to generate a completely discrete approximation for a large class of continuous time, continuous state stochastic systems. The crucial idea of this paper is that this method not only provides state estimates but also generates a rich approximation for the whole dynamical system along with it. The algorithms proposed here can generate this approximation in an incremental fashion and hence are amenable to practical applications. These algorithms were applied to the nonlinear optimal filtering problem and experiments show that they compare favorably to the state of the art. There are straightforward applications to a number of related state-estimation problems such as smoothing and MAP decoding, the latter we demonstrate in this paper. Directions for future work include creating discrete approximations of POMDPs that can be solved relatively easily to converge to an optimal policy for the general continuous time POMDP in an incremental manner.

References

[1] S.J. Julier and J.K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, volume 3, page 26. Citeseer, 1997.

[2] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *Radar and Signal Processing, IEE Proceedings F*, volume 140, pages 107–113. IET, 1993.

[3] A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Springer Verlag, 2001.

[4] R. Douc and O. Cappé. Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 64–69. Ieee, 2005.

[5] D. Fox. Adapting the sample size in particle filters through KLD-sampling. *The International Journal of Robotics Research*, 22(12):985, 2003.

[6] M.K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, pages 590–599, 1999.

[7] D. Crisan. Particle filters in a continuous time framework. In *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pages 73–78. IEEE, 2006.

[8] T. Lyons and N. Victoir. Cubature on wiener space. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 460(2041):169, 2004.

[9] P. Del Moral and L. Miclo. Branching and interacting particle systems approximations of Feynman-Kac formulae with applications to non-linear filtering. *Seminaire de Probabilites XXXIV*, pages 1–145, 2000.

[10] H.J. Kushner. *Probability methods for approximations in stochastic control and for elliptic equations*, volume 129. New York: Academic Press, 1977.

[11] H. J. Kushner and P. Dupuis. *Numerical methods for stochastic control problems in continuous time*. Springer Verlag, 2001.

[12] H.J. Kushner. Numerical approximations to optimal nonlinear filters. 2008.

[13] J.H. Reif. Complexity of the mover's problem and generalizations extended abstract. In *Proceedings of the 20th Annual IEEE Conference on Foundations of Computer Science*, pages 421–427, 1979.

[14] L Kavraki and J C Latombe. Probabilistic Roadmaps for Robot Path Planning. 1998.

[15] S.M. LaValle and J.J. Kuffner Jr. Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5):378–400, 2001.

[16] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *International Journal of Robotics Research*, 30(7):846–894, 2011.

[17] A. Perez, S. Karaman, M. Walter, A. Shkolnik, E. Frazzoli, and S. Teller. Asymptotically-optimal path planning for manipulation using incremental sampling-based algorithms. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[18] J. Jeon, S. Karaman, and E. Frazzoli. Anytime computation of time-optimal off-road vehicle maneuvers using the RRT*. In *IEEE Conference on Decision and Control (CDC)*, 2011.

[19] V. Huynh, S. Karaman, and E. Frazzoli. An incremental sampling-based algorithm for stochastic optimal control. In *IEEE Conference on Robotics and Automation*, 2012 (to appear).

[20] Bernt Karsten Øksendal. *Stochastic differential equations*. an introduction with applications. Springer Verlag, 2003.

[21] H. J. Kushner. Dynamical equations for optimal nonlinear filtering. *Journal of Differential Equations*, 3:179–190, 1967.

[22] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. Intelligent robotics and autonomous agents. MIT Press, 2005.

[23] H. Samet. Applications of spatial data structures. 1990.

[24] H. Risken. *The Fokker-Planck equation: Methods of solution and applications*, volume 18. Springer Verlag, 1996.

[25] H. J. Kushner and A. S. Budhiraja. A Nonlinear Filtering Algorithm Based on an Approximation of the Conditional Distribution. *IEEE Transactions on Automatic Control*, 45(3):580–585, 2000.

[26] A. Doucet and V.B. Tadić. Parameter estimation in general state-space models using particle methods. *Annals of the institute of Statistical Mathematics*, 55(2):409–422, 2003.

[27] G.D. Forney Jr. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

Preprint submitted to 51st IEEE Conference on Decision and Control .
Received March 7, 2012.