Sampling-based algorithm for filtering using Markov chain approximations

Sampling-based Algorithm for Filtering using Markov chain Approximations

Pratik Chaudhari Sertac Karaman Emilio Frazzoli

***Abstract*— In this paper, the filtering problem for a large class of continuous-time, continuous-state stochastic dynamical systems is considered. Inspired by recent advances in asymptotically-optimal sampling-based motion planning algorithms, such as the PRM* and the RRT*, an incremental sampling-based algorithm is proposed. Using incremental sampling, this approach constructs a sequence of Markov chain approximations, and solves the filtering problem, in an incremental manner, on these discrete approximations. It is shown that the trajectories of the Markov chain approximations converge in distribution to the trajectories of the original stochastic system; moreover, the optimal filter calculated on these Markov chains converges to the optimal continuous-time nonlinear filter. The convergence results are verified in a number of simulation examples.**

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 optimal state estimation of linear systems for decades now. Nonlinear filtering methods like Extended Kalman filter, and the Unscented Kalman filter have been introduced to expand its applicability.

In many applications, however, nonlinear dynamics and non-Gaussian noise prohibit closed-form expressions for the optimal filter. The seminal paper by Gordon, Salmond and Smith [1] introduced the bootstrap filter, which forms the basis for a class of general filters known as sequential Monte-Carlo methods [2]. They utilize a large number of random samples (called "particles") to represent arbitrary posterior distributions, and are propagated in time using importance sampling techniques. Moreover, sampling techniques, such as posterior resampling [3] and adaptive sampling [4], allow addressing a wide class of nonlinear and non-Gaussian models. However, for robust execution, it is necessary to tune the filtering algorithm for the problem at hand (see, e.g., [5]). In fact, arguably, the crucial aspect of particle filtering is estimating a good posterior to sample from.

A large amount of attention has been devoted also to continuous-time filtering algorithms, such as the Kalman-Bucy filter for continuous-time linear systems with additive Gaussian noise. More recent results include continuous-time particle filters that are inspired by weak approximations of solutions of stochastic differential equations (SDEs) [6]. Moreover, nonlinear filtering algorithms have been constructed using numerical solutions to the partial differential equations arising from the Zakai equation and the Kushner-Stratonovich equation. However, applications of this class of algorithms have been limited due to computational intractability or non-recursive nature which prevents them from being used on real-world online applications.

In this paper, we focus on the continuous-time nonlinear filtering problem. Our main results draw inspiration from two main areas. Firstly, we are interested in the Markov chain approximation approach, developed by Kushner [7], which has been proposed as a method to generate discrete approximations of ordinary SDEs with continuous representations of states, controls, and observations. The method is applicable to a wide class of problems in optimal stochastic control [7] and optimal estimation [8], but its applications in the literature are scarce, arguably due to the computational complexity incurred by "a priori" discretization of high-dimensional state spaces.

In a different context, the curse of dimensionality has been shown to be inevitable in almost all robot motion planning problems. In particular, the motion planning problem, i.e., the problem of finding a dynamically feasible trajectory around obstacles so as to reach a goal region is shown to be PSPACE-hard [9]. Yet, algorithms with probabilistic guarantees such as Probabilistic RoadMaps (PRM) [10] or the Rapidly-exploring Random Trees (RRT) [11] have been shown to work effectively in returning a feasible solution in high-dimensional configuration spaces, relying on effective discretization based on sampling. Most recently, in [12], two novel motion planning algorithms were proposed, namely the PRM* and the RRT*, which guarantee asymptotic optimality, i.e., almost-sure convergence to optimal trajectories, without sacrificing computational efficiency. In particular, the RRT* algorithm has been successfully applied to many challenging motion planning problems involving high-dimensional configuration spaces [13], complex dynamical systems [14]. Similar incremental sampling methods were also used to tackle stochastic optimal control problems [15].

We leverage these results to construct incrementally refined Markov chain approximations of stochastic systems, and incrementally solve the filtering problem on these discretizations. The resulting algorithms inherit many features of other sampling-based algorithms. In particular, they 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 new platforms with different computational capabilities.

This paper is organized as follows. The continuous-time nonlinear filtering problem for a stochastic dynamical system

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

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. Results of computational experiments are presented in Section VII.

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 the SDE

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

where (i) $x(t) \in \mathbb{R}^d$, (ii) $f : \mathbb{R}^d \to \mathbb{R}^d$, $F : \mathbb{R}^d \to \mathbb{R}^{d\times k}$ are Borel-measurable functions, (iii) $\{w(t) : t \geq 0\}$ is the standard k-dimensional Brownian motion, and 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 [16]. We tacitly assume throughout that $f(\cdot)$ and $F(\cdot)$ are bounded and continuous, to guarantee weak existence and weak uniqueness for the solutions of Equation (1).

In the nonlinear filtering problem [7], [16], 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

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

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 an l-dimensional Brownian motion independent of $\{x(t) : t \geq 0\}$. Similarly, we assume that $g(\cdot)$ and $G(\cdot)$ are bounded and continuous to guarantee weak existence and weak uniqueness of solutions to Equation (2). As in [7], we formulate the problem so that the system evolves inside a compact set, $\mathcal{S} \subset \mathbb{R}^d$. The process is stopped, if it hits the boundary of $\mathcal{S}$, i.e., at time $\tau := \inf\{s : x(s) \notin \mathcal{S}^o\}$, where $\mathcal{S}^o$ denotes the interior of $\mathcal{S}$. The filtering problem can be formally defined as follows.

Problem 1 *Given a set $\mathcal{Y}_t := \{y(s) : s \leq t\}$ of observations generated by the process in Equation* (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, based on observations $\mathcal{Y}_t$, the error-minimizing state estimate $\hat{x}(t)$ 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 [16]. In some references the filtering problem is posed as the estimation of the distribution of the random variable $\mathbb{E}[x(t) \,|\, H_t]$ (see, e.g., [7]), which is equivalent to our formulation of the problem. Let us also note that the solution of the filtering problem can be given by the following formula [7]. 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

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

where $R(t) = \exp\left[\int_0^t g(\tilde{x}(s))^T dy(s) - \frac{1}{2}\int_0^t |g(\tilde{x}(s))|^2 ds\right]$. Roughly speaking, in our approach, the Markov chain approximation method generates a process $\tilde{x}(t)$ that has the same law as the original process $x(t)$; Then, Equation (3), a limiting version of Bayes' rule, gives the optimal filter.

III. Preliminaries

A. Optimal Filter on a Markov chain

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'. Given an initial state $z_0 \in S$, let $\{\xi_i; i \in \mathbb{N}\}$ denote the (discrete-time) trajectory of the Markov chain $\mathcal{M}$ starting from z_0.

The optimal filtering problem for a discrete Markov chain is similar to Problem 1. Given a Markov chain $\mathcal{M} = (S, P)$ and a set $Y_k = \{y_i : i = 1, 2, \ldots, k\}$ of 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 $\phi_k(z) = \mathbb{P}(\xi_k = z \,|\, Y_k)$ as $\phi_n(z) = \sum_{z'\in S} \mathbb{P}(\xi_n = z, \xi_{n-1} = z' \,|\, Y_n)$, 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 except for the fact that observations come from an observation space instead of a finite set and the same formulae hold. Also note that the observations in Equation (2) are often discretized at times $k\delta$ as $y_k = g(x(k\delta))\delta + G(x(k\delta))[v(k\delta) - v(k\delta - \delta)]$, or alternatively,

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

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 the observation noise $\tilde{v}_k$ need not be Gaussian in our formulation.

B. The Markov Chain Approximation Method

Let $\mathcal{M} = (S, P, \Delta t)$ be a tuple where (S, P) is a Markov chain as defined above and $\Delta t : S \to \mathbb{R}_{>0}$ associates 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. Henceforth, we call this tuple a Markov chain for convenience. $\xi(\cdot)$ is the *continuous-time interpolation* of a discrete trajectory $\{\xi_i;\ i \in \mathbb{N}\}$ under holding times Δt, i.e., $\xi(\tau) = \xi_i \quad \text{for all } \tau \in [t_i, t_{i+1})$, where $t_i = \sum_{j=1}^{i} \Delta t(\xi_j)$.

Let $\{\mathcal{M}_n; n \in \mathbb{N}\}$, where $\mathcal{M}_n = (S_n, P_n, \Delta t_n)$, 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* [7] with the original system described by Equation (1) if the following criteria are satisfied *for all* $z \in S$.

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

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

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

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

Theorem 2 (Theorem 10.4.1 in [7]) *If $\{\mathcal{M}_n;\ n \in \mathbb{N}\}$ is a sequence of Markov chains, locally consistent with the stochastic dynamical system given by Equation* (1)*, $\xi_n(\cdot)$ has a subsequence that converges in distribution to $x(\cdot)$ such that*

$$x(t) = x_0 + \int_0^t f(x(s))ds + \int_0^t F(x(s))\ dw(s), \tag{8}$$

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

IV. Construction of Approximating 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 within a distance $r = \gamma\ (\log n/n)^{1/d}$ from z, i.e.,

$$\texttt{Near}(z, S) = \left\{ z' \in S,\ z' \neq z : \|z' - z\|_2 \leq \gamma \left(\frac{\log n}{n}\right)^{1/d} \right\}$$

where $n = |S|$, $d = \dim(S)$, and $\gamma > 0$ is a constant that will be specified in Section VI.

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 $\Delta t = \frac{\texttt{distance}}{\texttt{average velocity}}$.

d) Transition Probabilities: Local consistency conditions are a set of linear equations for transition probabilities. However, we can also use an approximation to get the probabilities as follows. Given a state $z \in \mathcal{S}$ and a finite set $Z_{\text{near}} \subset \mathcal{S}$, the $\texttt{ComputeTransProb}(z, Z_{\text{near}}, \Delta t(z))$ procedure returns a function $p(\cdot \mid z)$ 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 $p(z' \mid z) = \eta\ \mathcal{N}_{\mu,\Sigma}(z')$ where $\mu = z + f(z)\Delta t(z)$ and $\Sigma = F(z)F(z)^T\Delta t(z)$ and the constant η ensures $\sum_{z' \in Z_{\text{near}}} p(z' \mid z) = 1$. Lemma 3 in Section VI proves that this satisfies local consistency conditions in the limit.

Algorithm 1: Incremental Markov chain Construction

```
1  n ← 0;
2  while n < N do
3      z ← Sample();
4      S_n ← S_{n-1} ∪ {z};
5      ConnectState(z, (S_n, P_{n-1}, Δt_{n-1}));
6      P_n ← P_{n-1},  Δt_n ← Δt_{n-1};
7      Z_near ← Near(z, S_n);
8      for z_near ∈ Z_near do
9          ConnectState(z_near, (S_n, P_n, Δt_n));
10     n ← n + 1;
11 return (S_N, P_N, Δt_N);
```

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

```
1  Δt(z) ← ComputeHoldingTime(z, S);
2  Z_near ← Near(z, S);
3  P(· | z) ← ComputeTransProb(z, Z_near, Δt(z));
```

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. We prove in Section VI that trajectories of an incremental construction of a Markov chain approximation converge in distribution to those of the original stochastic system.

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 only 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.

D. Computational complexity

Without providing the technical details, we sketch the computational complexity of the proposed algorithms. The

Algorithm 3: Batch Markov chain construction

```
1  n ← 0;
2  while n < N do
3      z ← Sample();
4      S ← S ∪ {z};
5      n ← n + 1;
6  for z ∈ S_N do
7      ConnectState(z, (S_N, P_N, Δt_N));
8  return (S_N, P_N, Δt_N);
```

`Near` procedure takes worst case $\mathcal{O}(\log n)$ time using approximate nearest neighbor algorithms [17]. Note that the expected number of samples in a ball of radius $\gamma\,(\log n/n)^{1/d}$ is $\mathcal{O}(\log n)$. The complexity of `ConnectState` is thus $\mathcal{O}(\log n)$ and it executes for an expected $\mathcal{O}(\log n)$ samples in lines 7-9 of Algorithm 1. 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 before-hand as in Algorithm 3.

V. Filtering on Markov chain approximations

Inorder to use Equation (4) to propagate estimates on the Markov chain constructed in Section IV, we require single step transition probabilities, i.e., roughly, the holding times of all states need to be the same.

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 `ComputeTransProbTime`$(z, Z_{\text{near}}, \Delta t, \delta)$ that returns a probability density function over $T_\delta \times Z_{\text{near}}$, where $T_\delta = \{0,\, \delta,\, 2\delta,\, \ldots,\}$. This probability density is denoted by $p^\delta(\cdot \,|\, k\delta, z)$ defined for $z \in S$ and $k \in \{1, 2, \ldots\}$. Let $p(\cdot \,|\, z) =$ `ComputeTransProb`$(z, Z_{\text{near}}, \Delta t(z))$ be computed as described in Section IV-A. $p^\delta(\cdot \,|\, k\delta, z)$ is constructed from $p(\cdot \,|\, z)$ as,

$$\begin{aligned} 1 - p^\delta\big(\, k\delta + \delta, z \,|\, k\delta, z\,\big) &= \delta/\Delta t(z) \\ \frac{p^\delta\big(\, k\delta, z \,|\, k\delta, z'\,\big)}{1 - p^\delta\big(\, k\delta + \delta, z \,|\, k\delta, z\,\big)} &= p(z \,|\, z') \end{aligned} \tag{9}$$

Equations (9) also satisfy local consistency conditions and thus the modified Markov chain can be used in Equation 4.

The condition $\delta \le \min_{z \in S_n} \Delta t(z)$ is required to ensure $p^\delta(\cdot, z \,|\, \cdot, z) \le 1$ for all states 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 (9). 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 that has $\Delta t(z_{n+1}) \le \delta_{\text{current}}$. Since $\delta \sim \Delta t(z) = \mathcal{O}((\frac{\log n}{n})^{2/d})$ from Section IV-A, two successive values of n, n_1, and n_2, when we have to recalculate the probabilities are exponentially increasing i.e., $n_2 \sim n_1 2^{d/2}$ giving 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 dependend only on the state, not on any particular time. Hence, it is not necessary to compute the transition probabilities for each time in T_δ separately in an implementation. The filtered estimate calculated using Equation (4) converges to the optimal estimate as $\delta \to 0$ and $n \to \infty$, for which we provide a proof in Section VI.

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 `ComputeTransProb` *procedure satisfies local consistency conditions given in Equations* (6) *and* (7)*.*

Proof: The Fokker-Plank equation [18] is a partial differential equation that describes the evolution of the probability density, $\phi(x(t))$ of the state of Equation (1),

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

Small time solution is given as, $P(x', t + \Delta t \,|\, x, t) = \frac{1}{\sqrt{2\pi F(x) F^T(x) \Delta t}}\, \exp\left(-\frac{1}{2}\frac{[x' - x - f(x,t)\Delta t]^2}{F(x)F^T(x)\Delta t}\right)$ The Gaussian approximation in `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 [12], i.e., the small time solution converges to the actual solution in the limit. ■

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

Proof: 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. 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 $k \in K_n$, $\mathbb{P}\left(A_{n,k}^c\right) = \left(1 - \frac{(\frac{\gamma}{2})^{-d}}{\mu(\mathcal{S})}\,\frac{\log n}{n}\right)^n \le \exp\left(-((\frac{\gamma}{2})^d/\mu(\mathcal{S}))\,\log n\right) = n^{-(\frac{\gamma}{2})^d/\mu(\mathcal{S})}$, where $\mu(\cdot)$ denotes the usual Lebesgue measure. Let A_n denote the event that all cells $G_n(i)$ contain at least one vertex of S_n. Then,

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

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

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

which is summable for $\gamma > 2\ (2\,\mu(\mathcal{S}))^{1/d}$. 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 $\gamma(\log n/n)^{1/d}$, every state z is connected to at least one other state.

This ensures that Equations (6) and (7) are satisfied for all samples $z \in S_n$. 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: Equations (6) and (7) are satisfied for Algorithm 1 by Theorem 4. 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 any iteration, say i, goes to zero. $\Delta t(z)$ is a decreasing function of n. We thus essentially prove that lines 7-9 are executed on every state z infinitely often.

Fix an iteration i and a 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 $\sum_{n=i+1}^{\infty} \mathbb{P}(A_n) = \infty$ and the event A_n is independent from A_i for all $i \neq n$, Borel-Cantelli lemma implies $\mathbb{P}(\limsup_{n\to\infty} A_n) = 1$. Hence, any state z is reconnected infinitely often, with probability one. ■

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 [8]) *Let $\phi(\cdot)$ be any continuous real-valued function, $x(t)$ be defined by Equation* (8) *and observations be given by $y(t)$. If $\xi_n(\cdot)$ is a sequence which converges in distribution as $n \to \infty$ to $x(t)$ and is independent of $(x(t), y(t))$,*

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

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

Consider the 2-dimensional stochastic dynamical system

$$\begin{aligned} dx_1 &= -\frac{1}{2}x_1\ dt + 0.03\ dw_1 \\ dx_2 &= -x_2\ dt + 0.03\ dw_2 \end{aligned} \tag{10}$$

In Figure 1, results from computational experiments involving $50,000$ simulated trajectories of the Markov chain $(x_{\text{Markov}}(t))$ and the original stochastic system $(x(t))$ are shown. A scatter plot for the state of the system is shown at five different time instants. Figure 2 shows the convergence of error in moments calculated over the same set of trajectories with the number of states ranging from $1,000$ to $100,000$. The experimental results presented in these figures verify the theoretical results presented in Theorems 4 and 5.

B. Filtering

In this section, we compare the proposed filtering algorithm with others such as the EKF and the particle filter.

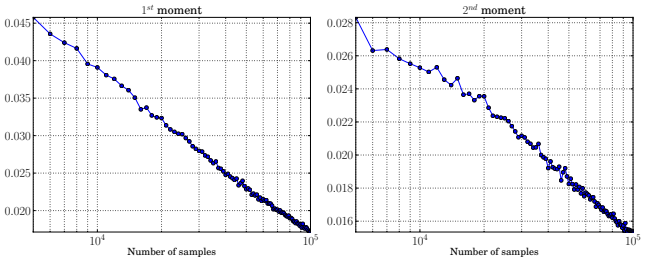
1) Drifting ship: Consider a ship [19] confined to move within a disc of radius 9 units. A large force $f_i(x(t))$ acts on the ship to make it move inwards if it is moving outwards when it goes out of this disc as shown in Equation (11). The ship is a 2-dimensional double integrator with forces $f_1(x)$, $f_2(x)$. Dynamics of all states is assumed to have noise of standard deviation e with observations being range and heading as given in Equation (12). Figure 3 shows that the tracking error is similar to that of the particle filter.

$$f_i(x(t)) = \frac{-50x_i}{\sqrt{x_1^2 + x_2^2}}\ \mathcal{I}_{\{\sqrt{x_1^2+x_2^2}\geq 9\}} \mathcal{I}_{\{x_1x_3+x_2x_4\geq 0\}} \tag{11}$$

$$\begin{aligned} dy_1 &= [x_1^2 + x_2^2]^{1/2}dt + e_1\ dv_1 \\ dy_2 &= \tan^{-1}(x_2/x_1)dt + e_2\ dv_2 \end{aligned} \tag{12}$$

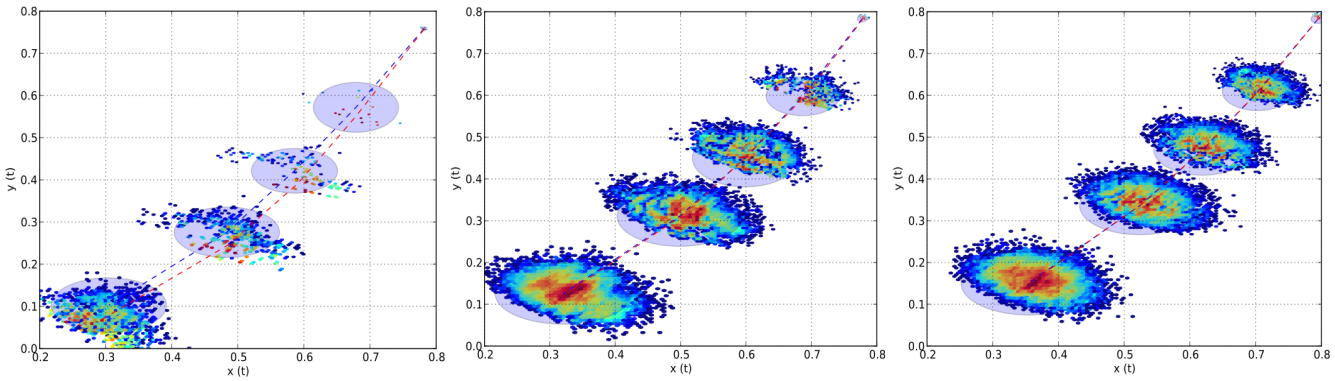
2) Van der Pol oscillator: Consider a noisy Van der Pol oscillator given by Equation (13). This system is highly nonlinear with a stable limit cycle for $\mu > 0$.

$$\begin{aligned} dx_1 &= x_2\ dt + e_1\ dw_1 \\ dx_2 &= [-x_1 + \mu\ x_1\ (1 - {x_1}^2)]\ dt + e_2\ dw_2 \\ dy &= x_1\ dt + e_3\ dv \end{aligned} \tag{13}$$



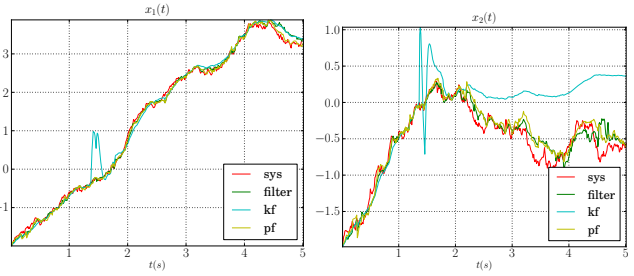
(a) (b)

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., $\|\mathbb{E}[\xi_n(T)\xi_n^T(T) - x(T)x^T(T)]\|_2$.



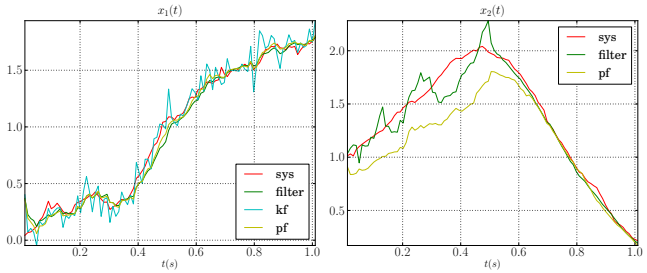
(a) 1000 samples (b) 10, 000 samples (c) 40, 000 samples

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 original stochastic system as given in Equation (10). The dotted blue and red lines show the mean of the original dynamics 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 is also seen to converge. 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 (11) with $e = 0.3$, $e_1 = 0.03$ and $e_2 = 0.03$. EKF diverges near $(0, 0)$ due to large nonlinearity in bearing observations. 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 last equation is the scalar observation equation and $\mu = 2$. Figure 4 shows the performance of the proposed algorithm on this system. Note that this system is typically hard for the EKF which accumulates linearization error due to varying time scales and, predictably, the EKF estimate of $x_2(t)$ 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 quality of estimate and as also computationally tractability.

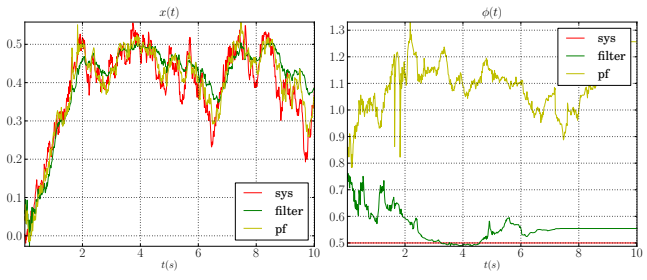
3) Parameter estimation: We compared these filters on a modified version of a parameter estimation problem from [20] as given in Equation (14). The parameter we are estimating is $\phi = 0.5$.

$$\begin{aligned} dx &= x\cos(2\pi\phi x)\ dt + \sigma_x\ dw_1 \\ d\phi &= 0\ dt + \sigma_\phi\ dw_2 \\ dy &= x\ dt + \sigma_v\ dv. \end{aligned} \tag{14}$$

To begin with, it is only known that $\phi \sim \mathcal{N}(0.8, 1)$. This is a hard problem for a particle filter because the conditional density of ϕ given data is not in the exponential family [20] 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 error.

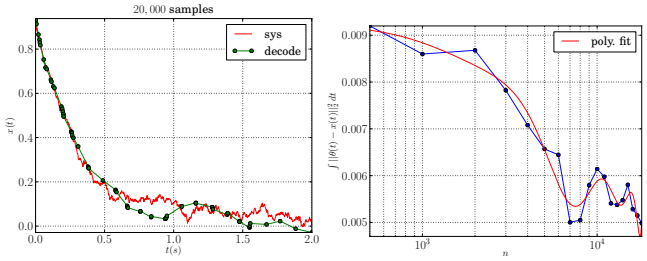
C. Implementation details

The algorithms proposed here are general and can be used to get a discrete approximation of a large class of stochastic systems. We have demonstrated an application of this idea



(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.



(a) Example decoded trajectory (b) Error convergence

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 4.9×10^{-3} with $20,000$ samples in Figure (a). The convergence of decoding error averaged over 100 trajectories is shown on a log-log plot in Figure (b). The red curve shows a 10^{th} order polynomial fit for the error data.

to optimal nonlinear filtering. Heuristics specific to the filtering problem applied to our Markov chain construction can vastly improve the computational complexity in practice. Roughly, uniform sampling of the state-space results in the convergence rate depending upon the size of state-space. We can avoid this by concentrating samples around the estimated posterior while creating the Markov chain. In the examples given in Section VII-B, propagating mean and variance of an assumed Gaussian prior enables an efficient online algorithm.

Next, we consider an application to MAP decoding, where we directly use the incremental Markov chain construction.

D. Maximum a posteriori (MAP) trajectory

In this section, we focus on the discrete-time MAP trajectory estimation (decoding) problem. 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., $\hat{x}_t = \arg\max_\xi \mathbb{P}(\xi_t \,|\, Y_t)$ (see [21] for a more elaborate formulation). A Markov chain $\mathcal{M}_n^\delta = (S_n, P_n, \delta)$ constructed using Algorithm 1 can be used with the Viterbi algorithm to get the most probable trajectory after t observations. The observation probability is given by $\mathbb{P}(y_k \,|\, z)$ 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.

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 filtering problem and experiments show that they compare favorably to the state of the art. Incremental solutions to related estimation problems such as smoothing (Forward-Backward algorithm) and trajectory decoding (Viterbi algorithm) can be obtained easily using the algorithms proposed here. Directions for future work include creating discrete approximations of partially observable Markov decision process (POMDPs) that can be solved in an incremental manner to converge to an optimal solution of the general continuous time POMDP.

Acknowledgments

This work was supported in part by the Army Research Office MURI grant W911NF-11-1-0046.

References

[1] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings–F*, 140(2):107–113, 1993.

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

[3] R. Douc and O. Cappé. Comparison of resampling schemes for particle filtering. In *Proc. Int. Symp. on Image and Signal Processing and Analysis*, pages 64–69, 2005.

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

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

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

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

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

[9] 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.

[10] L.E. Kavraki, P. Svestka, J.C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

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

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

[13] 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.

[14] 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.

[15] 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).

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

[17] H. Samet. *Applications of spatial data structures*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1990.

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

[19] 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.

[20] 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.

[21] O. Zeitouni and A. Dembo. A maximum a posteriori estimator for trajectories of diffusion processes. *Stochastics: An International Journal of Probability and Stochastic Processes*, 20(3):221–246, 1987.