When is Particle Filtering Efficient for Planning in Partially Observed Linear Dynamical Systems?

Simon S. Du[1] **Wei Hu**[2] **Zhiyuan Li**[2] **Ruoqi Shen**[1] **Zhao Song**[2] **Jiajun Wu**[3]

[1]University of Washington
[2]Princeton University
[3]Stanford University

Abstract

Particle filtering is a popular method for inferring latent states in stochastic dynamical systems, whose theoretical properties have been well studied in machine learning and statistics communities. In many control problems, e.g., partially observed linear dynamical systems (POLDS), oftentimes the inferred latent state is further used for planning at each step. This paper initiates a rigorous study on the efficiency of particle filtering for sequential planning, and gives the first particle complexity bounds. Though errors in past actions may affect the future, we are able to bound the number of particles needed so that the long-run reward of the policy based on particle filtering is close to that based on exact inference. In particular, we show that, in stable systems, polynomially many particles suffice. Key in our proof is a coupling of the ideal sequence based on the exact planning and the sequence generated by approximate planning based on particle filtering. We believe this technique can be useful in other sequential decision-making problems.

1 INTRODUCTION

Many real-world applications require planning on a partially observed stochastic dynamic system [Kaelbling et al., 1998]. The planning policy often operates on the underlying latent states instead of directly on raw observations. Take robot navigation as an example. The raw observations are high-dimensional RGB-D videos, and it is often preferred to instead plan upon the underlying latent state, such as the location of the robot.

A core challenge is to infer these latent states from observations. For simple stochastic systems such as hidden Markov models (HMM) corrupted by a Gaussian noise, there are analytical solutions for inference, i.e., Kalman filtering [Kalman, 1960]. However, exact inference is often computationally infeasible in many stochastic systems with complex probabilistic models. A typical example is inferring the latent state of a partially observable linear dynamical system (POLDS), especially in recent models that parametrize the transition and emission probability distributions with deep neural networks [Hausknecht and Stone, 2015]. Here, while exact computation of the transition kernel and the stochastic emission kernel is efficient (the same computation complexity as using deep neural networks for prediction), exact computation of the posterior distribution over latent states is infeasible.

Particle filtering or Sequential Monte Carlo is a generic approach to *approximately* infer the underlying latent states in stochastic dynamical systems (cf. Algorithm 1). Instead of computing the posterior distribution exactly, this approach simulates a set of particles according to the transition kernel. Then, a weighted average of the particles is used to approximate the posterior distribution, where the weight of each particle is given by its likelihood. Particle filtering is computationally efficient because it only needs to compute the transition kernel and the stochastic emission kernel, but does not require computing the posterior distribution.

Particle filtering as approximate inference of latent states can be naturally integrated with *belief space planning* [Platt Jr et al., 2010]. Recently, researchers have also proposed various approximations to make the steps within particle filtering differentiable, so that the inference networks can be trained end-to-end with policy networks [Karkus et al., 2017, 2018a,b, Jonschkowski et al., 2018, Wang et al., 2019]. In terms of applications, however, these works mostly focus on visual navigation, where the planning horizon is short (with instant feedback), and the reward function varies smoothly and continuously with respect to actions such as moving forward. Applications in dynamic systems without these properties are rare. This gives rise to a theoretical question:

Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI 2021), PMLR 161:728–737.

Is particle filtering *provably efficient* for sequential planning on stochastic systems?

While the theory of particle filtering for inference is well-studied in statistical machine learning, the theory of particle filtering for planning is rather unexplored. The approximation error in inference can lead to selection of different actions and further affect the outcome such as cumulative rewards in the future. Therefore, we not only need to study the approximation in the inference, but also how the error affects the future planning.

In this paper, we initiate the rigorous quantitative study to characterize the efficiency of particle filtering in terms of the properties of stochastic system. We study the fundamental hidden Markov model, where the dynamics of transition and emission are linear, but the noise in transition and emission can be arbitrary probabilistic distributions. We focus on the planning problem in which we assume noise distributions are known. Unless these noise distributions are within specific classes such as Gaussian, exact inference for latent states is computationally infeasible, and approximate inference such as particle filtering is needed. Our analysis not only applies to popular linear, time-invariant (LTI) systems, but also time-varying ones. It can also potentially be extended for nonlinear dynamic systems with recent development in Koopman analysis [Brunton et al., 2016], which will enable additional applications in robotic control [Bruder et al., 2019].

Our Contribution Our main contribution is an upper bound on the number of particles needed for particle filtering–based planning to be close to the planning based on the exact inference. To our knowledge, this is the first non-asymptotic theoretical result on particle filtering for sequential planning. The bound depends on some control-theoretic quantities that describe the POLDS, the planning horizon, the Lipschitzness of the reward function, and the inverse of the likelihood of observations and the target sub-optimality. We also complement the upper bound with a lower bound showing the dependency is necessary.

Main Challenge and Analysis Overview The main challenge in the analysis is studying the distribution of the particles. When there is no sequential planning, the particles are generated independently, so their distributions can be easily studied. However, when doing sequential planning, i.e., when the states of the particles depend on the past actions, the particles are not independent anymore. The actions taken is based on the particle approximation of the past states, so the particles are correlated with each other. To avoid analyzing the complicated joint distribution of the particles directly, we show that it is enough to analyze the particle approximation of the noise in each round separately. The simulated noise in each particle is independent and can be easily analyzed.

To study the performance of particle filtering–based sequential planning, we need to compare the approximate process generated by particle filtering with an ideal process generated by exact inference. To make sure that the two processes can be fairly compared, we couple the approximate sequence with the ideal sequence using the same noise. It can be hard to compare those two processes because after taking different actions, the two processes are not estimating the same state anymore. In the following time steps, the two processes will take actions based on estimations of different states. Then, even when there is no estimation error present, two processes can still grow further apart. In this paper, we show that although the error can accumulate and be amplified through actions in sequential planning, we can still upper bound the number of particles needed so that the long-run rewards of the two processes remain close. We believe our framework can be the starting point of future study on particle filtering for sequential planning and can be useful for studying other methods.

Organization This paper is organized as follows. In Section 1.1, we discuss related works. In Section 2, we introduce necessary notations and formally state the problem. In Section 3, we present our main result, an upper bound on the particle complexity of particle filtering – based sequential planning. In Section 4, we complement the upper bound with a lower bound. In Section 5, we conclude and list concrete open problems. In Appendix, we show the proofs that are omitted in the main paper and present some simulation studies.

1.1 RELATED WORK

Here we discuss related theoretical work.

For inference, the quality of particle filtering is often measured by the distance between the posterior from the exact inference and that from the approximate inference, in metrics such as L_2 distance and Kullback-Leibler divergence. Many works have analyzed the number of particles needed to make the distance small, conditioned on properties of the dynamic system [Whiteley et al., 2016, Huggins et al., 2019, Crisan and Doucet, 2002, Marion and Schmidler, 2018, Chopin et al., 2004, Oreshkin et al., 2011]. However, to our knowledege, no prior work analyzed the quality of particle filtering–based planning. In this setting, the distance between the posterior from exact inference and approximate inference is not sufficient to measure for quality of particle filtering, as the approximation error in inference can lead to the selection of a different action and in turn affect the total reward.

Controlling a known dynamical system is a classical problem [Bertsekas et al., 1995]. For the POLDS setting considered in this paper, the controller often needs to first infer the latent state and plan on top of it. When the noise distribution is Gaussian and the reward is quadratic, the problem becomes Linear-Qudratic-Gaussian (LQG) control. For LQG, one can first use analytical formulas for inference [Kalman, 1960], and then, by the separation principle, directly apply an linear controller on the inferred latent state. Unfortunately, for most noise distributions, exact inference is in general computationally intractable and we must resort to approximate inference techniques such as particle filtering. Recently, researchers also try to leverage online learning techniques to design provable algorithms for control with known and unknown linear dynamical systems [Agarwal et al., 2019a,b, Li et al., 2019, Cohen et al., 2018, Even-Dar et al., 2009, Goel and Hassibi, 2020, Yu et al., 2009, Abbasi-Yadkori et al., 2014, Neu and Gómez, 2017, Foster and Simchowitz, 2020, Dean et al., 2019, Tsiamis et al., 2020, Hazan et al., 2017, Simchowitz et al., 2020, Simchowitz, 2020], some of which can be even applied to adversarial noise. Our work differs from this line of research as particle filtering–based planning is a fundamentally different approach.

2 PRELIMINARIES

Notations For any positive integer n, we use $[n]$ to denote the set of integers $\{1, 2, \cdots, n\}$. For vector x, we use $\|x\|$ to denote its ℓ_2 norm. For matrix A, we use $\|A\|_{op}$ denote the operator norm of A. For time t, we use $x_{0:t}$ to denote the sequence $x_0, ..., x_t$. For event E, we use $\Pr[E]$ to denote the probability that the event E happens. For random variable X, Y and their realizations x, y, we use $\mathbb{P}_X[x]$ to denote the value of the probability density function of X at x and $\mathbb{P}_X[x|y]$ to denote the value of the density function of X conditional on $Y = y$ at x. We write $\mathbb{P}_X[x]$ and $\mathbb{P}_X[x|y]$ as $\mathbb{P}[x]$ and $\mathbb{P}[x|y]$ when there is no ambiguity. For any function f, we use $\widetilde{O}(f)$ to denote the class $O(f) \cdot \log^{O(1)}(f)$.

Problem Setup We study the setting of planning in POLDS. At each time step $t = 1, ..., T$, the environment is in some latent state $x_t \in \mathcal{X} \subseteq \mathbb{R}^d$. The agent receives a partial observation $o_t \in \mathcal{O} \subseteq \mathbb{R}^m$ of the latent state x_t and takes action $\widehat{u}_t \in \mathcal{U} \subseteq \mathbb{R}^k$ based on the observations in the current time step and previous time steps, $o_{0:t}$. The action causes the environment to change to the new state x_{t+1} based on a known transition kernel. Finally, in time step T, we receive a reward R, which is a function of the past states and actions.

To put it formally, in our setting, at $t = 0$, we start from a known state x_0, which can be observed exactly. For $t = 0, ..., T-1$, we have the state updated as

$$x_{t+1} = A_t \cdot x_t + B_t \cdot \widehat{u}_t + \xi_t. \tag{1}$$

$\widehat{u}_t$ is the action we take at time t. $A_t \in \mathbb{R}^{d\times d}$ and $B_t \in \mathbb{R}^{d\times k}$ are transition matrices on the state x_t and the action $\widehat{u}_t$, respectively, at time t. We assume that the matrices A_t and B_t are known. $\xi_t \in \mathbb{R}^d$ is some transition noise following a known distribution $\xi_t \sim \mu_t(\cdot)$. The action $\widehat{u}_t$ is taken based on the observations $o_{1:t}$ of the current state and the past states.

Algorithm 1 Particle Filtering for Sequential Planning

```
Input: starting state $x_0$, number of particles $N$, number of time steps $T$.
for $i = 1 \to N$ do
    Initialize particle weight $w_0^{(i)} = 1$.
    Initialize particle state $x_0^{(i)} = x_0$.
end for
for $t = 0 \to T-1$ do
    Estimate latent state $\widehat{y}_t \leftarrow \frac{\sum_{i=1}^N w_t^{(i)} x_t^{(i)}}{\sum_{i=1}^N w_t^{(i)}}$.
    Take action $\widehat{u}_t = g(\widehat{y}_t)$ and observe $o_{t+1}$.
    for $i = 1 \to N$ do
        Generate random noise $\xi_t^{(i)} \sim \mu_t(\cdot)$.
        Update particle state $x_{t+1}^{(i)} = A_t x_t^{(i)} + B_t \widehat{u}_t + \xi_t^{(i)}$.
        Update particle weight $w_{t+1}^{(i)} \leftarrow w_t^{(i)} \cdot \eta_{t+1}(o_{t+1} - C_{t+1} x_{t+1}^{(i)})$.
    end for
end for
```

At state x_t, the observation o_t is given by

$$o_t = C_t \cdot x_t + \zeta_t. \tag{2}$$

where $C_t \in \mathbb{R}^{m\times d}$ is a known transition matrix and ζ_t is some noise following a known distribution $\eta_t(\cdot)$.

In our setting, we are given a policy $g : \mathcal{X} \to \mathcal{U}$, which is a function on the state space. However, since we only have access to a partial observation of the latent state x_t, we can only infer the state x_t based on the observations $o_1, ..., o_t$. We use particle filtering to do the latent state inference, which is listed in Algorithm 1.

Particle filtering estimates the latent state by simulating a group of particles using the known transition kernel. Those particles update their states using the same actions as we take in the real process. Then, the state estimation is given by a weighted average of the simulated states of the particles. The weight of each particle is proportional to the likelihood of the states of that particle given the observations.

Formally, we simulate N particles, $x_{0:T}^{(1)}, ..., x_{0:T}^{(N)} \in \mathcal{X}^T$. All N particles start from the same starting state x_0. In time step t, the particles are updated according to

$$x_{t+1}^{(i)} = A_t \cdot x_t^{(i)} + B_t \cdot \widehat{u}_t + \xi_t^{(i)}. \tag{3}$$

The action $\widehat{u}_t$ is the same as the action taken in step t of the real process $x_{0:T}$. $\xi_t^{(i)}$ are sampled independently according to the noise distribution $\mu_t(\cdot)$.

Next, we show how, at time t, we use the simulated particles $x_{0:t}^{(1)}, ..., x_{0:t}^{(N)}$ to estimate the latent state x_t. The weight of particle i at time $t > 0$, $w_t^{(i)}$, is given by

$$w_t^{(i)} = \prod_{s=1}^{t} \mathbb{P}\left[o_s \mid x_s^{(i)}\right] = \prod_{s=1}^{t} \eta_s \left(o_s - C_s \cdot x_s^{(i)}\right).$$

The weight $w_t^{(i)}$ of the i-th particle measures how likely the true latent states, $x_{0:t}$, are the states of the particle i, $x_{0:t}^{(i)}$, given the observations $o_{1:t}$. We give higher weights to particles with more likely states. Then, the estimated state $\widehat{y}_t$ is a weighted average of the states of the particles,

$$\widehat{y}_t = \frac{\sum_{i=1}^{N} w_t^{(i)} x_t^{(i)}}{\sum_{i=1}^{N} w_t^{(i)}}.$$

We note that if an infinite amount of particles are simulated, the estimated state would be the posterior mean of the state given the observations. Given the estimated state $\widehat{y}_t$, we take action $\widehat{u}_t$ to be

$$\widehat{u}_t = g(\widehat{y}_t).$$

Note we study policies that only depend on the estimated hidden state, which follows the separation principle in stochastic control theory. This class of policies is optimal for certain settings [Bertsekas et al., 1995].

To study the efficiency of the particle filtering algorithm, we compare the approximate process, $x_{0:t}$, described above, with an ideal process, $x_{0:t}^*$, which are generated via exact inference. The ideal process starts from $x_0^* = x_0$. For $t = 0, ..., T-1$, the ideal process is updated as

$$x_{t+1}^* = A_t \cdot x_t^* + B_t \cdot u_t^* + \xi_t, \tag{4}$$

The action u_t^* is taken based on exact inference, which will be defined formally later. Similarly, the observation o_t^* for $t = 1, ..., T$, is generated according to

$$o_t^* = C_t \cdot x_t^* + \zeta_t, \tag{5}$$

The transition matrices A_t, B_t and C_t are the same as those used in generating the approximate process. To make sure that the two processes can be fairly compared, we let the transition noise ξ_t and the observation noise ζ_t in the ideal process be the same as those in the approximate process.

Now, we show how the action u_t^* is chosen in the ideal process. We assume that in the ideal process, we can compute the exact posterior mean of the state x_t^* given the observations $o_{1:t}^*$. We estimate the state x_t^* as

$$\begin{aligned}\widetilde{y}_t =& \frac{\int_{x'_{1:t} \in \mathcal{X}^t} \prod_{s=1}^{t} \mathbb{P}\left[o_s^* \mid x'_s\right] x'_t \mathrm{d}\rho_t(x'_{1:t})}{\int_{x'_{1:t} \in \mathcal{X}^t} \prod_{s=1}^{t} \mathbb{P}\left[o_s^* \mid x'_s\right] \mathrm{d}\rho_t(x'_{1:t})} \\ =& \frac{\int_{x'_{1:t} \in \mathcal{X}^t} \prod_{s=1}^{t} \eta_s\left(o_s^* - C_s \cdot x'_s\right) x'_t \mathrm{d}\rho_t(x'_{1:t})}{\int_{x'_{1:t} \in \mathcal{X}^t} \prod_{s=1}^{t} \eta_s\left(o_s^* - C_s \cdot x'_s\right) \mathrm{d}\rho_t(x'_{1:t})}.\end{aligned}$$

where ρ_t is the distribution of $x'_{1,t}$ given actions $u_{0:t-1}^*$ and starting state x_0. Given the estimation $\widetilde{y}_t$, the action u_t^* is taken to be $u_t^* = g(\widetilde{y}_t)$.

In this paper, we study how accurately particle filtering can approximate the exact inference and how the error of particle filtering can affect the long-run reward where the reward function $r_T : \mathcal{X}^T \times \mathcal{U}^T \to \mathbb{R}$ maps some states $x_{1:T}$ and actions $u_{0:T-1}$ in the past T time steps to a reward value in $\mathbb{R}$. In particular, we study the number of particles needed so that the reward at time step T, $r_T(x_{1:t}, \widehat{u}_{0:T-1})$, of the approximate process is close to that of the ideal process, $r_T(x_{1:t}^*, u_{0:T-1}^*)$.

3 MAIN RESULTS

In this section, we present our main theoretical results. First, we introduce some necessary regularity conditions. Next, we discuss our results for general non-linear policies. Lastly, we focus on linear policies and give more refined results.

3.1 REGULARITY ASSUMPTIONS

To formally state our results, we first describe our assumptions.

Assumption 3.1. *The transition noise $\xi \in \mathbb{R}^d$ is sub-Gaussian with parameter $1/m$, i.e.,*

$$\mathbb{E}\left[e^{u^\top(\xi - \mathbb{E}\,\xi)}\right] \le e^{\|u\|^2/(2m)}, \textit{ for any vector } u \in \mathbb{R}^d.$$

Assumption 3.1 is standard regularity condition on transition noise. Without a regularity condition, the noise can arbitrarily large.

Assumption 3.2. *The reward function r_T is L_r-Lipschitz, i.e.,*

$$\begin{aligned}&|r_T(x_{1:t}, u_{0:t-1}) - r_T(x'_{1:t}, u'_{0:t-1})| \\ &\le\ L_r \cdot \left(\sum_{t=1}^{T} \|x_t - x'_t\| + \sum_{t=0}^{T-1} \|u_t - u'_t\|\right),\end{aligned}$$

for all $x_{1:T}, x'_{1:T} \in \mathcal{X}^T$ and $u_{0:T-1}, u'_{0:T-1} \in \mathcal{U}^T$.

Assumption 3.2 is regularity condition imposed on the reward condition. Note since we are considering planning based on approximate inference, we cannot hope to choose the same action as the one based on the exact inference. Similarly, we cannot hope to have same sequence of hidden states based on particle filtering as that based on the exact inference. Assumption 3.2 bounds how much small deviation on the action and hidden state sequence will affect the reward.

3.2 MAIN RESULT FOR GENERAL NON-LINEAR POLICIES

Now we discuss our result on non-linear policies. We need some assumptions about the dynamical system and the policy to characterize the stability of the system. Such assumptions are necessary for control problems. For the non-linear policy, we consider the following assumption.

Assumption 3.3. *The policy g is L_g-Lipschitz. i.e., for all $x, x' \in \mathcal{X}$, $\|g(x)-g(x')\| \le L_g \cdot \|x-x'\|$,for all $0 \le t_1 < t_2 < T$, $\left\|\Pi_{s=t_1}^{t_2} A_s\right\|_{op} \le C_a \rho_a^{t_2-t_1}$, and for all $0 \le t < T$, $\|B_t\|_{op} \le C_b$ for some $L_g, C_a, \rho_a, C_b > 0$.*

Assumption 3.3 imposes a Lipschitz condition on the policy g. Note for this assumption, the policy g can be non-linear. The condition $\left\|\Pi_{s=t_1}^{t_2} A_s\right\|_{op} \le C_a \rho_a^{t_2-t_1}$ describes the growth rate of the dynamical system. Such assumption is standard in the control literature. The bound on $\|B_t\|_{op}$ ensures a small deviation on the action will not alter the system by much.

Our main result is the following.

Theorem 3.4. *Given any accuracy $\epsilon \in (0, 1/2)$, failure probability $\delta > 0$ and number of time steps $T \ge 1$. Under Assumptions 3.1, 3.2 and 3.3, let*

$$\Sigma_a^{(T)} = 1 + C_a \sum_{s=0}^{T-2} \rho_a^s \text{ and } \Sigma_{ab}^{(T-1)} = \sum_{s=0}^{T-2} (C_a + C_b L_g)^s.$$

Let

$$\Delta_T = L_r L_g \Sigma_a^{(T)} \left(1 + C_b \Sigma_a^{(T)}\right) \left(1 + L_g C_b \Sigma_{ab}^{(T-1)}\right)$$

and

$$p = \mathbb{P}_{O_{1:T}}\left[o_{1:T} \mid \widehat{u}_{0:t-1}, x_0\right].$$

For any $\delta > 0$, it is enough to use

$$N = \widetilde{O}(T^2 \Delta_T^2 d m^{-1} \epsilon^{-2} p^{-1})$$

particles so that with probability at least $1-\delta$,

$$|r_T(x_{1:T}, \widehat{u}_{0:T-1}) - r_T(x^*_{1:T}, u^*_{0:T-1})| \le \epsilon.$$

Theorem 3.4 shows as long as the number of particles scales *polynomially* with p, parameters in Assumptions 3.1 and 3.2 and a quantity Δ_T defined by the parameters in Assumption 3.3, the reward collected by particle filtering–based planning is close to that of the ideal process. Here, p is the likelihood of the observations $o_{1:T}$ conditional on the initial state and the actions. When the space of the observation is discrete, p is the probability of seeing the observations. We note that in some cases, it is possible that $1/p$ grows exponentially as the number of time steps T grows. However, we are able to show it is necessary for the number of particles to depend on $1/p$. The lower bound on the dependence on $1/p$ is stated in Section 4. To our knowledge, this is the first non-asymptotic particle complexity analysis for planning problems.

To prove the theorem, we first show the number of particles needed so that the particle can approximate the latent state, especially the transition noise, accurately. Then, we show how the error in each time step can accumulate through the planning process. We show the proof ideas in Section 3.4 and defer the complete proof to Appendix.

Our bound depends on Δ_T which in turn depends on two quantities $\Sigma_a^{(T)}$ and $\Sigma_{ab}^{(T-1)}$ which together describe the growth rate of the system, i.e., how stable the system is. To better illustrate Theorem 3.4, we consider the benign scenario where the system is stable in the sense that $\rho_a \le 1$ and $C_a + C_b Lg \le 1$. Stable systems are widely studied in the control literature. The following corollary shows if the system is stable, then the number of particles only needs to scale polynomially with all parameters.

Corollary 3.5. *In the same setup as Theorem 3.4, suppose $\rho_a \le 1$ and $C_a + C_b L_g \le 1$. Then it is enough to use*

$$N = \widetilde{O}(T^6 d m^{-1} L_r^2 L_g^2 (1 + C_b^2 T^2) \epsilon^{-2} p^{-1})$$

particles so that for any $\delta > 0$, with probability at least $1-\delta$, $|r_T(x_{1:T}, \widehat{u}_{0:T-1}) - r_T(x^*_{1:T}, u^*_{0:T-1})| \le \epsilon$.

3.3 MAIN RESULT FOR LINEAR POLICIES

In this section, we focus on the linear policy, i.e., $g(x) = Gx$ for some matrix G. Linear policy is a popular class and is widely studied in the control and online learning literature. We make the following assumption about the policy and the system.

Assumption 3.6. *$\|G\|_{op} \le L_g$, for all $0 \le t_1 < t_2 < T$, $\left\|\Pi_{s=t_1}^{t_2} (A_s + B_s G)\right\|_{op} \le C_{ab} \rho_{ab}^{t_2-t_1}$, $\left\|\Pi_{s=t_1}^{t_2} A_s\right\|_{op} \le C_a \rho_a^{t_2-t_1}$, and for all $0 \le t < T$, $\|B_t\|_{op} \le C_b$, and $\|B_t G\|_{op} \le C_{bg}$ for some $L_g, C_{ab}, \rho_{ab}, C_a, \rho_a, C_b, C_{bg} > 0$.*

Assumption 3.6 can be viewed as a fine-grained version of Assumption 3.2. Recall that Theorem 3.4 depends on $(C_a + C_b L_g)$ which corresponds to the condition $\left\|\Pi_{s=t_1}^{t_2} (A_s + B_s G)\right\|_{op} \le C_{ab} \rho_{ab}^{t_2-t_1}$. Note $(C_a + C_b L_g)^{t_2-t_1}$ is always an upper bound of $\left\|\Pi_{s=t_1}^{t_2} (A_s + B_s G)\right\|_{op}$, so the condition $\left\|\Pi_{s=t_1}^{t_2} (A_s + B_s G)\right\|_{op} \le C_{ab} \rho_{ab}^{t_2-t_1}$ is a more refined characterization. We remark that this condition is also a common one in the control literature. Similarly, Theorem 3.4 depends on $L_g C_b$, which corresponds to the condition $\|B_t G\|_{op} \le C_{bg}$ in Assumption 3.6. C_{bg} is a more refined characterization of $\|B_t G\|_{op}$ than $L_g C_b$. Now we present our general theoretical result.

Theorem 3.7. *For any accuracy $\epsilon \in (0, 1/2)$, failure probability $\delta > 0$ and number of time steps $T \geq 1$. Under Assumption 3.1, 3.2 and 3.6, let $\Sigma_a^{(T)} = 1 + C_a \sum_{s=0}^{T-2} \rho_a^s$ and $\bar{\Sigma}_{ab}^{(T-1)} = 1 + C_{ab} \sum_{s=0}^{T-3} \rho_{ab}^s$. Let $\Delta_T = L_r L_g \Sigma_a^{(T)} \left(1 + C_b \Sigma_a^{(T)}\right) \left(1 + C_{bg} \bar{\Sigma}_{ab}^{(T-1)}\right)$. For any $\delta > 0$, it is enough to use*

$$N = \widetilde{O}(T^2 \Delta_T^2 d m^{-1} \epsilon^{-2} p^{-1})$$

particles so that with probability at least $1 - \delta$,

$$|r_T(x_{1:T}, \widehat{u}_{0:T-1}) - r_T(x^*_{1:T}, u^*_{0:T-1})| \leq \epsilon.$$

Similar to Theorem 3.4, Theorem 3.7 also guarantees that with a sufficiently large number of particles, the reward collected by particle filtering–based planning is close to that of the ideal process. The main difference is that Theorem 3.7 depends on parameters in Assumption 3.6, which are finer-grained characterizations of the process. This requires some new proof components that exploit the linearity of the policy.

Again, to better illustrate our result, we provide the following corollary for the stable system.

Corollary 3.8. *In the same setup as Theorem 3.4, suppose $\rho_a \leq 1$ and $\rho_{ab} \leq 1$. Then it is enough to use*

$$\begin{aligned} N = \widetilde{O}\big(&T^2 d m^{-1} L_r^2 L_g^2 \epsilon^{-2} p^{-1} (1 + C_a^2 T^2) \\ &(1 + C_b^2 + C_b^2 C_a^2 T^2)(1 + C_{bg}^2 + C_{bg}^2 C_{ab}^2)\big) \end{aligned}$$

particles so that for any $\delta > 0$, with probability at least $1 - \delta$,

$$|r_T(x_{1:T}, \widehat{u}_{0:T-1}) - r_T(x^*_{1:T}, u^*_{0:T-1})| \leq \epsilon.$$

The conditions $\rho_a \leq 1, \rho_{ab} \leq 1$ appeared in many studies on linear systems. Corollary 3.8 guarantees that for these systems, the number of particles only needs to scale polynomially with all parameters.

3.4 PROOF OF MAIN RESULTS

In this section, we state the proof idea of our main results, Theorem 3.4 and Theorem 3.7. We defer a complete version of the proof to Appendix.

We first note that at time t, since we know the initial state x_0, the transition matrices $A_{0:t-1}$ and $B_{0:t-1}$ and the past actions $\widehat{u}_0, ..., \widehat{u}_{t-1}$, estimating the state x_t is equivalent to estimating $\xi_0, ..., \xi_{t-1}$.

Lemma 3.9. *For any $t \in [T]$, we can write the state x_t as*

$$x_t = \sum_{s=0}^{t-1} \prod_{s'=s+1}^{t-1} A_{s'} \cdot (\xi_s + B_s \cdot \widehat{u}_s) + \prod_{s=0}^{t-1} A_s \cdot x_0, \quad (6)$$

the state x^*_t as

$$x^*_t = \sum_{s=0}^{t-1} \prod_{s'=s+1}^{t-1} A_{s'} \cdot (\xi_s + B_s \cdot u^*_s) + \prod_{s=0}^{t-1} A_s \cdot x_0, \quad (7)$$

and for particle $i \in [N]$,

$$x_t^{(i)} = \sum_{s=0}^{t-1} \prod_{s'=s+1}^{t-1} A_{s'} \cdot \left(\xi_s^{(i)} + B_s \cdot \widehat{u}_s\right) + \prod_{s=0}^{t-1} A_s \cdot x_0. \quad (8)$$

Proof. (6) follows directly from applying the definition of the process given in (1) recursively. Similarly, (7) and (8) can be obtained by the definitions (4) and (3). □

Recall from Section 2 that the estimation $\widehat{y}_t$ is given by a weighted average of the states of the simulated particles,

$$\widehat{y}_t = \frac{\sum_{i=1}^N w_t^{(i)} x_t^{(i)}}{\sum_{i=1}^N w_t^{(i)}}. \quad (9)$$

and the estimation $\widetilde{y}_t$ is given by the posterior mean of x^*_t given observations $o_{0:t}$,

$$\widetilde{y}_t = \frac{\int_{x'_{1:t} \in \mathcal{X}^t} \prod_{s=1}^t \mathbb{P}\left[o^*_s \mid x'_s\right] x'_t \mathrm{d}\rho_t(x'_{1:t})}{\int_{x'_{1:t} \in \mathcal{X}^t} \prod_{s=1}^t \mathbb{P}\left[o^*_s \mid x'_s\right] \mathrm{d}\rho_t(x'_{1:t})}. \quad (10)$$

By Lemma 3.9, we know that to estimate x_t and x^*_t, it is enough to estimate $\xi_{0:t-1}$. We can further show that the estimators $\widehat{y}_t$ and $\widetilde{y}_t$ can be written as a function of estimators $\widehat{\xi}_{t,0:t-1}$ and $\widetilde{\xi}_{t,0:t-1}$, past actions $\widehat{u}_{0:t-1}$ and $u^*_{0:t-1}$, and the initial state x_0. The estimator $\widehat{\xi}_{t,0:t-1}$ is given by a weighted average of the noise of the particles, $\xi^{(1)}_{0:t-1}, ..., \xi^{(N)}_{0:t-1}$, similar to (9). The estimator $\widetilde{\xi}_{t,0:t-1}$ is given by the posterior mean of the noise given observations, similar to (10). Since $\widehat{u}_{0:t-1}$ and $u_{0:t-1}$ are determined by $\widehat{y}_{0:t-1}$ and $\widetilde{y}_{t-1}$, to show that $\widehat{y}_t$ is close to $\widetilde{y}_t$, it is enough to show that $\widehat{\xi}_{s,0:s-1}$ is close to $\widetilde{\xi}_{s,0:s-1}$ in all rounds $s = 1, ..., t-1$. We can show the following concentration bound.

Lemma 3.10 (Particle Concentration). *Let $M := \sqrt{\frac{d}{m}(1 + 2\sqrt{\log \beta'/d} + 2 \log \beta'/d)}$ for some $\beta' > 1$. At time $t \in [T]$, for each $s = 0, .., t-1$, we have for any $\beta \leq \frac{1}{2}$,*

$$\|\widehat{\xi}_{t,s} - \widetilde{\xi}_{t,s}\| \leq 4\beta M,$$

holds with probability at least

$$1 - (d+1)\exp(-N\beta^2\gamma_t/3) - N\exp(-\beta').$$

Lemma 3.10 studied how the particle approximation concentrates in one time step. We next discuss how the error of approximation in one time step can affect the actions in the future and further affect the long-run reward of the process.

From Lemma 3.9, it is easy to see that the distance between x_t and x_t^* is determined by the distance between actions in the past time steps, $\widehat{u}_{0:t-1}$ and $u_{0:t-1}^*$. It is easy to see that the distance between x_t and x_t^* is determined by the distance between actions in the past time steps, $\widehat{u}_{0:t-1}$ and $u_{0:t-1}^*$.

Lemma 3.11. *At time t,*

$$x_t - x_t^* = \sum_{s=0}^{t-1}\left(\prod_{s'=s+1}^{t-1} A_{s'}\right) B_s\left(\widehat{u}_s - u_s^*\right).$$

Proof. The proof follows directly from the problem setup. □

Then, to bound the distance between the states x_t and x_t^*, it suffices to bound the distance between the actions $\widehat{u}_{0:t-1}$ and $u_{0:t-1}^*$. We show the bound in Lemma 3.12.

Lemma 3.12. *Assume that* $\max_{0\le s<t\le T}\|\widehat{\xi}_{t,s} - \widetilde{\xi}_{t,s}\| = \epsilon$. *At time t, we can show the following bounds on* $\|\widehat{u}_t - u_t^*\|$.

- *Under Assumptions 3.1, 3.2 and 3.3, for* $t \in [T]$, *let* $\Sigma_a^{(t)} = 1 + C_a\sum_{s=0}^{t-2}\rho_a^s$ *and* $\Sigma_{ab}^{(t-1)} = \sum_{s=0}^{t-2}(C_a + C_bL_g)^s$. *Then, we have*

$$\|\widehat{u}_t - u_t^*\| \le L_g\Sigma_a^{(t)}\left(1 + L_gC_b\Sigma_{ab}^{(t-1)}\right)\cdot\epsilon.$$

- *Under Assumptions 3.1, 3.2 and 3.6, for* $t \in [T]$, *let* $\Sigma_a^{(t)} = 1 + C_a\sum_{s=0}^{t-2}\rho_a^s$ *and* $\bar{\Sigma}_{ab}^{(t-1)} = 1 + C_{ab}\sum_{s=0}^{t-3}\rho_{ab}^s$. *Then, we have*

$$\|\widehat{u}_t - u_t^*\| \le L_g\Sigma_a^{(t)}\left(1 + C_{bg}\bar{\Sigma}_{ab}^{(t-1)}\right)\cdot\epsilon.$$

Finally, we use Lemma 3.12 to prove our main results Theorem 3.4 and Theorem 3.7.

Proof of Theorem 3.4 and Theorem 3.7. We state the proof for the Lipschitz g case here. The proof for linear g follows the same steps. We first show the number of particles needed so that the estimation of the noise, ξ_t, in a single round is accurate. If

$$N = \Omega(\beta^{-2}p^{-1}\log(dT/\delta)), \tag{11}$$

then

$$\begin{aligned}(d+1)\cdot\exp(-N\beta^2\gamma_t/3) &\le (d+1)\cdot\exp(-N\beta^2 p/3)\\ &\le \delta/(2T^2),\end{aligned}$$

where the first inequality follows from

$$\gamma_t = \mathbb{P}_{O_{1:t}^*}[o_{1:t}^*|u_{0:t}^*, x_0] = \mathbb{P}_{O_{1:t}}[o_{1:t}|\widehat{u}_{0:t}, x_0] \ge p.$$

Let $M := \sqrt{\frac{d}{m}(1 + 2\sqrt{\log\beta'/d} + 2\log\beta'/d)}$. If we choose $\beta' = \log(2T^2N/\delta)$ and $\beta = \epsilon/(4MT)$, by Lemma 3.10, with success probability at least

$$1 - \sum_{t=1}^{T}\sum_{s=0}^{t-1}\delta/(2T^2) - \sum_{t=1}^{T}\sum_{s=0}^{t-1}\delta/(2T^2) \ge 1-\delta,$$

we have for all time step $t = 1, ..., T$ and $s = 0, .., t-1$,

$$\|\widehat{\xi}_{t,s} - \widetilde{\xi}_{t,s}\| \le 4\beta M = \epsilon/T.$$

Next, we bound the distance between actions in the two processes. By Lemma 3.12, for any $t = 1, ..., T$,

$$\|\widehat{u}_t - u_t^*\| \le L_g\Sigma_a^{(t)}\left(1 + L_gC_b\Sigma_{ab}^{(t-1)}\right)\cdot\frac{\epsilon}{T}. \tag{12}$$

The second inequality follows from our assumption. By Lemma 3.11 and our assumptions, we can further bound the distance between the states of the two processes as

$$\begin{aligned}\|x_t - x_t^*\| &= \left\|\sum_{s=0}^{t-1}\left(\prod_{s'=s+1}^{t-1}A_{s'}\right)B_s(\widehat{u}_s - u_s^*)\right\|\\ &\le C_b\left(\|\widehat{u}_{t-1} - u_{t-1}^*\| + C_a\sum_{s=0}^{t-2}\rho_a^s\|\widehat{u}_s - u_s^*\|\right)\\ &\le C_b\Sigma_a^{(t)}\cdot L_g\Sigma_a^{(t)}\left(1 + L_gC_b\Sigma_{ab}^{(t-1)}\right)\cdot\frac{\epsilon}{T},\end{aligned} \tag{13}$$

Thus, combining (12) and (13), we can get for any L_r-Lipschitz reward function r_T,

$$\begin{aligned}& r_T(x_{1:T}, \widehat{u}_{0:T-1}) - r_T(x_{1:T}^*, u_{0:T-1}^*)\\ \le & \sum_{t=1}^{T}L_r\|x_t - x_t^*\| + \sum_{t=1}^{T}L_r\|\widehat{u}_{t-1} - u_{t-1}^*\|\\ \le & L_rL_g\Sigma_a^{(T)}\left(1 + C_b\Sigma_a^{(T)}\right)\left(1 + L_gC_b\Sigma_{ab}^{(T-1)}\right)\epsilon,\end{aligned}$$

where the first step follows from r_T is L_r-Lipschitz and the second step follows from (12) and (13).

Plugging $\beta^2 = \epsilon^2/(16T^2M^2) = \widetilde{\Theta}(\epsilon^2T^{-2}d^{-1}m)$ into (11), the number of particles needed is

$$N = \widetilde{O}(\beta^{-2}p^{-1}) = \widetilde{O}(T^2dm^{-1}\epsilon^{-2}p^{-1}),$$

which completes the proof. Similarly, we can also show that the number of particles needed for linear g so that

$$\begin{aligned}& r_T(x_{1:T}, \widehat{u}_{0:T-1}) - r_T(x_{1:T}^*, u_{0:T-1}^*)\\ \le & L_rL_g\Sigma_a^{(T)}\left(1 + C_b\Sigma_a^{(T)}\right)\left(1 + C_{bg}\bar{\Sigma}_{ab}^{(T-1)}\right)\epsilon\end{aligned}$$

is

$$N = \widetilde{O}(T^2dm^{-1}\epsilon^{-2}p^{-1}).$$

□

4 LOWER BOUND

In this section, we show that Algorithm 1 has particle complexity with at least a linear dependence on the inverse of the likelihood of observation, $1/p$. We note that it is possible for $1/p$ to depend exponentially on the number of time step T in some processes. However, we are able to show that it is necessary for the particle complexity to depend on $1/p$. Precisely, in Theorem 3.7, we show that we need $O(1/p)$ particles to approximate the whole process well. We show in this section that the upper bound $O(1/p)$ is tight.

We consider the following process of dimension $d = 1$. We start from the initial state $x_0 = 0$. The total number of time steps is T. At each time step $t = 0, ..., T-1$, let the transition matrices $A_t = 1$ and $B_t = 0$.

Let $\boldsymbol{\delta}(\cdot)$ be the standard Dirac Delta function such that $\boldsymbol{\delta}(x-a) = 0$ for $x \neq a$ and $\int_{a-\epsilon}^{a+\epsilon} \boldsymbol{\delta}(x-a)\mathrm{d}x = 1$ for $\epsilon > 0$. Then, let the density function of the transformation noise ξ_t be given by

$$\mu_t(\xi) = \frac{1}{2}\boldsymbol{\delta}(\xi - 1) + \frac{1}{2}\boldsymbol{\delta}(\xi + 1).$$

for $t = 0, ..., T-1$. Finally, for all time steps $t = 1, ..., T$, let the observation matrix $C_t = 1$ and the observation noise ζ_t be always 0, i.e., $\eta_t(\zeta) = \boldsymbol{\delta}(\zeta)$.

Now, we consider the observation $o_T = T$. From the way we construct the process, it is clear that $x_t = t$ for all $t = 1, ..., T$. Then, we must have $\xi_t = 1$ for all $t = 0, ..., T-1$. Moreover, for the observation $o_T = T$, $p = \mathbb{P}[o_{1:T}|x_0] = 2^{-T}$. We state this formally in Lemma 4.1.

Lemma 4.1. *If $o_T = T$, then $x_t = t$ for all $t = 1, ..., T$ and therefore $\xi_t = 1$ for all $t = 0, ..., T-1$. Moreover, for the observation $o_T = T$, $p = \mathbb{P}[o_{1:T}|x_0] = 2^{-T}$.*

Proof of Lemma 4.1. Since $\zeta_1 = ... = \zeta_T = 0$, $x_t = o_t$ for all $t = 1, ..., T$.

Assume for contradiction that there exists some $\xi_s < 0$ for some $0 < s < T$. Then we have

$$x_T \leq T - 1 < T,$$

contradicting $x_T = T$. Thus, we have $x_t = o_t = t$ for all $t = 1, ..., T$. Moreover,

$$\begin{aligned} p &= \int_{\xi'_{0:T-1}} \mathbb{P}[o_{1:T}|\xi'_{0:T-1}, x_0]\mathrm{d}\pi_T(\xi'_{0:T-1}) \\ &= \int_{\xi'_{0:T-1}} \prod_{t=0}^{T-1} \eta_{t+1}\left(t + 1 - \sum_{s=0}^{t} \xi_s\right) \mathrm{d}\pi_T(\xi'_{0:T-1}) \\ &= \Pr[\xi_t > 0, \forall 0 \leq t \leq T-1] \\ &= 2^{-T}. \end{aligned}$$

The second step follows from the definition of our process. The third step follows from

$$\prod_{t=0}^{T-1} \eta_{t+1}\left(t + 1 - \sum_{s=0}^{t} \xi_s\right) = 0$$

if there exists some $\xi_t < 0$. The last step follows from $\xi_t > 0$ with probability $1/2$, □

Next, we show that if we do not simulate enough particles, then with high probability, there will not exist any particle $i \in [N]$ that has $\xi_t^{(i)} > 0$ for all $t = 1, ..., T$. Then, all particles will have weight $w_T^{(i)} = 0$ at time step T.

Theorem 4.2. *Suppose the number of simulated particles $N \leq 1/(2kp)$ for some $k > 1$. For $i \in [N]$, let I_i be the indicator random variable of the event that $\xi_t^{(i)} > 0$ for all $t = 1, ..., T$. Then we have $\Pr\left[\sum_{i=1}^{N} I_i \geq 1\right] \leq \frac{1}{k}$.*

Proof of Theorem 4.2. Since for each $i \in [N]$, $I_i = 0$ with probability $1 - 2^{-T} = 1 - p$ and $I_i = 1$ with probability $2^{-T} = p$,

$$\Pr\left[\sum_{i=1}^{N} I_i \geq 1\right] \leq \sum_{i=1}^{N} \Pr\left[I_i = 1\right] = \frac{1}{k},$$

which completes the proof. □

By Theorem 4.2, to avoid all the weights of particle going to zero after T time steps with high probability, we need to simulate at least $\Omega(1/p)$ particles.

5 CONCLUSION

This paper gives the first quantitative analysis of using particle filtering for planning over latent states. We also demonstrate the conditions in our theorem are necessary. In the following, we list some open problems for future study.

Optimal Particle Complexity A natural interesting theoretical problem is, under the assumptions in Section 3: *what is the minimal number of particles needed to find a near-optimal planning policy?* Note the standard particle filtering algorithm (cf. Algorithm 1) is only one approach that uses particles. One can design more advanced algorithms that operate on these particles with smaller particle complexity. For example, particle filtering resampling has shown to outperform standard particle filtering algorithm [Kitagawa, 1993], and it is possible that this approach also admits theoretical benefits. On the other hand, proving particle complexity lower bound will also improve our understanding on methods based on particles in general. We believe designing an algorithm that achieve optimal particle complexity will have impact in both theory and practice.

Learning with Particle Filtering Our work assumes the probabilistic models of transition and emission are known. Recently, a line of work used particle filtering in both training and planning phases [Karkus et al., 2018a, Jonschkowski et al., 2018]. While we have analyzed the planning phase, the analysis for training the probabilistic models is more challenging. In this problem, one uses particle filtering to explore the state space and collect the data to train probabilistic models. Characterizing the sample and particle complexity together is an interesting direction to pursue.

Acknowledgements

Part of the work was done while SSD, WH, ZL, RS, ZS were participating the Optimization, Statistics, and Theoretical Machine Learning program at Institute for Advanced Study of Princeton. SSD was supported by NSF DMS-1638352 and the Infosys Membership. WH and ZL were supported by NSF, ONR, Simons Foundation, Schmidt Foundation, Amazon Research, DARPA and SRC. JW is supported by ARO MURI W911NF-15-1-0479, the Samsung Global Research Outreach Program, Amazon, IBM, and Stanford HAI.

References

Yasin Abbasi-Yadkori, Peter Bartlett, and Varun Kanade. Tracking adversarial targets. In *International Conference on Machine Learning*, pages 369–377, 2014.

Naman Agarwal, Brian Bullins, Elad Hazan, Sham M Kakade, and Karan Singh. Online control with adversarial disturbances. *arXiv preprint arXiv:1902.08721*, 2019a.

Naman Agarwal, Elad Hazan, and Karan Singh. Logarithmic regret for online control. In *Advances in Neural Information Processing Systems*, pages 10175–10184, 2019b.

Dimitri P Bertsekas, Dimitri P Bertsekas, Dimitri P Bertsekas, and Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995.

Daniel Bruder, Brent Gillespie, C David Remy, and Ram Vasudevan. Modeling and control of soft robots using the koopman operator and model predictive control. In *Robotics: Science and Systems (RSS)*, 2019.

Steven L Brunton, Bingni W Brunton, Joshua L Proctor, and J Nathan Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PloS one*, 11(2), 2016.

Nicolas Chopin et al. Central limit theorem for sequential monte carlo methods and its application to bayesian inference. *The Annals of Statistics*, 32(6):2385–2411, 2004.

Alon Cohen, Avinatan Hassidim, Tomer Koren, Nevena Lazic, Yishay Mansour, and Kunal Talwar. Online linear quadratic control. *arXiv preprint arXiv:1806.07104*, 2018.

Dan Crisan and Arnaud Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50(3):736–746, 2002.

Sarah Dean, Nikolai Matni, Benjamin Recht, and Vickie Ye. Robust guarantees for perception-based control. *arXiv preprint arXiv:1907.03680*, 2019.

Eyal Even-Dar, Sham M Kakade, and Yishay Mansour. Online markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

Dylan J Foster and Max Simchowitz. Logarithmic regret for adversarial online control. *arXiv preprint arXiv:2003.00189*, 2020.

Gautam Goel and Babak Hassibi. The power of linear controllers in lqr control. *arXiv preprint arXiv:2002.02574*, 2020.

Matthew Hausknecht and Peter Stone. Deep recurrent Q-learning for partially observable MDPs. In *2015 AAAI Fall Symposium Series*, 2015.

Elad Hazan, Karan Singh, and Cyril Zhang. Learning linear dynamical systems via spectral filtering. In *Advances in Neural Information Processing Systems*, pages 6702–6712, 2017.

Jonathan H Huggins, Daniel M Roy, et al. Sequential monte carlo as approximate sampling: bounds, adaptive resampling via ∞-ess, and an application to particle gibbs. *Bernoulli*, 25(1):584–622, 2019.

Rico Jonschkowski, Divyam Rastogi, and Oliver Brock. Differentiable particle filters: End-to-end learning with algorithmic priors. In *Robotics: Science and Systems (RSS)*, 2018.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

Peter Karkus, David Hsu, and Wee Sun Lee. Qmdp-net: Deep learning for planning under partial observability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Peter Karkus, David Hsu, and Wee Sun Lee. Integrating algorithmic planning and deep learning for partially observable navigation. *arXiv preprint arXiv:1807.06696*, 2018a.

Peter Karkus, David Hsu, and Wee Sun Lee. Particle filter networks with application to visual localization. In *Conference on Robot Learning (CoRL)*, 2018b.

Genshiro Kitagawa. Monte carlo filtering and smoothing method for non-gaussian nonlinear state space model. *Inst. Statist. Math. Res. Memo.*, 1993.

Yingying Li, Xin Chen, and Na Li. Online optimal control with linear dynamics and predictions: Algorithms and regret analysis. In *Advances in Neural Information Processing Systems*, pages 14858–14870, 2019.

Joseph Marion and Scott C Schmidler. Finite sample complexity of sequential monte carlo estimators. *arXiv preprint arXiv:1803.09365*, 2018.

Gergely Neu and Vicenç Gómez. Fast rates for online learning in linearly solvable markov decision processes. *arXiv preprint arXiv:1702.06341*, 2017.

Boris N Oreshkin, Mark J Coates, et al. Analysis of error propagation in particle filters with approximation. *The Annals of Applied Probability*, 21(6):2343–2378, 2011.

Robert Platt Jr, Russ Tedrake, Leslie Kaelbling, and Tomas Lozano-Perez. Belief space planning assuming maximum likelihood observations. In *Robotics: Science and Systems (RSS)*, 2010.

Max Simchowitz. Making non-stochastic control (almost) as easy as stochastic. *arXiv preprint arXiv:2006.05910*, 2020.

Max Simchowitz, Karan Singh, and Elad Hazan. Improper learning for non-stochastic control. *arXiv preprint arXiv:2001.09254*, 2020.

Anastasios Tsiamis, Nikolai Matni, and George Pappas. Sample complexity of kalman filtering for unknown systems. In *Learning for Dynamics and Control*, pages 435–444. PMLR, 2020.

Yunbo Wang, Bo Liu, Jiajun Wu, Yuke Zhu, Simon S Du, Li Fei-Fei, and Joshua B Tenenbaum. Dual sequential monte carlo: Tunneling filtering and planning in continuous POMDPs. *arXiv preprint arXiv:1909.13003*, 2019.

Nick Whiteley, Anthony Lee, Kari Heine, et al. On the role of interaction in sequential monte carlo algorithms. *Bernoulli*, 22(1):494–529, 2016.

Jia Yuan Yu, Shie Mannor, and Nahum Shimkin. Markov decision processes with arbitrary reward processes. *Mathematics of Operations Research*, 34(3):737–757, 2009.