Efficient State and Parameter Estimation of Nonlinear State-Space Models through Probabilistic Optimal Control

Victor Vantilborgh[⋆] VICTOR.VANTILBORGH@UGENT.BE
Mohammad M. Filabadi[⋆]
Guillaume Crevecoeur[⋆]
Tom Lefebvre[⋆]
[⋆] *Faculty of Engineering and Architecture, Ghent University, 9000 Ghent, Belgium*

Editors: G. Sukhatme, L. Lindemann, S. Tu, A. Wierman, N. Atanasov

Abstract

This work presents a novel representation of the smoothing distribution - the posterior state distribution of a discrete-time dynamical system - through its connection to probabilistic optimal control. The key idea is to represent the posterior as the closed-loop behavior of a synthetic control system governed by an optimal stochastic policy. This formulation enables forward simulation of equally weighted trajectories that capture the statistics of the posterior without the need for backward sampling or importance weighting. We derive a practical algorithm based on probabilistic dynamic programming to compute this policy efficiently, with linear computational complexity in the number of particles. Furthermore, the uniform particle weighting significantly simplifies and accelerates the Expectation–Maximization algorithm, providing substantial benefits for system identification of nonlinear dynamical systems with latent states. The proposed method offers a simple, stable, and scalable alternative to traditional particle smoothers and demonstrates accurate parameter estimation and model learning at significantly reduced computational cost.

1. Introduction

Nonlinear probabilistic state-space models (PSSMs), also known as Hidden Markov Models (HMMs), provide a principled framework for modeling stochastic dynamical systems and time-series data across a wide range of applications Murphy (2012); Särkkä (2013). In dynamic systems, the performance of model-based controllers and estimators strongly depends on the accuracy of the identified underlying model. A widely used approach for identifying such models is the Expectation–Maximization (EM) algorithm, which requires evaluating the posterior - or smoothing - distribution of the latent states Schön et al. (2015, 2018); Rosato et al. (2022). In the general nonlinear and non-Gaussian setting, this posterior smoothing density must be approximated - often from noisy, incomplete measurements and (partially) unknown governing system equations - making the computation both challenging and computationally demanding.

A popular class of methods for approximating the smoothing density are *particle smoothers*, which rely on Monte Carlo averaging over a set of weighted trajectories. Although conceptually straightforward, these methods typically exhibit quadratic computational complexity in the number of particles given a typical *forward-then-backward* processing structure. Moreover, particle degeneracy and the need for resampling complicate their use within iterative algorithms such as EM, particularly in system identification of nonlinear systems with latent states.

This work introduces an efficient alternative by reformulating the smoothing problem as a probabilistic optimal control problem. The posterior is represented as the closed-loop behavior of a synthetic control system governed by an optimal stochastic policy. This formulation enables forward simulation of equally weighted trajectories that capture the posterior statistics without backward passes or importance weighting. The proposed algorithm, derived via probabilistic dynamic programming, exhibits linear computational complexity in the number of particles and produces a uniformly weighted representation of the posterior. These properties make it particularly advantageous for system identification and model learning, as the uniform particle weights simplify and accelerate the EM algorithm while improving numerical stability.

The main contributions of this work are as follows:

- We reformulate the discrete-time Bayesian smoothing problem as a probabilistic optimal control problem.
- We derive the Ensemble Gradient Probabilistic Dynamic Programming Smoother (EG-PDPS), a computationally efficient algorithm with linear complexity in the number of particles that generates equally weighted Monte Carlo trajectories representing the smoothing distribution.
- We demonstrate the utility of the particle representation of the computed posterior density in a sample-based EM algorithm for nonlinear system identification, achieving accurate model learning at significantly reduced computational cost.

2. Problem formulation and motivation

2.1. Modelling and notation

Consider a probabilistic state-space model, $\mathcal{M}$, with state $x \in \mathcal{X} \subset \mathbb{R}^{n_x}$ and observations $z \in \mathcal{Z} \subset \mathbb{R}^{n_z}$ Särkkä (2013) with initial state density $x_0 \sim p(x_0|\square)$, transition density function $x_t \sim p_\theta(x_t|x_{t-1})$ and observation density function $z_t \sim p_\theta(z_t|x_t)$, with $\square$ representing the deterministic root node of the Markov Chain (i.e. we can start the Markov process deterministically in $\square$ at time $t = 0$). The subscript θ explicitly indicates that the representations of the density functions are parametrized by some parameter θ. To enhance readability, we will omit this notation throughout the paper, except where it is necessary to emphasize the parameterization.

The joint probability density is expressed as[1]

$$p(\underline{x}_T, \underline{z}_T) = p(x_0|\square)p(z_0|x_0) \prod_{t=1}^{T} p(z_t, x_t|x_{t-1}) \tag{1}$$

where density $p(x_t, z_t|x_{t-1})$ is defined as $p(x_t|x_{t-1})p(z_t|x_t)$. Note that conditioning on $\square$ is silent henceforth.

Remark 1 *A popular nonlinear instance of a PSSM is given by* $x_0 \sim \mathcal{N}(x_0|\mu_0, \Sigma_0)$, $x_{t+1} = f_t(x_t, q_t)$ *and* $z_t = g_t(x_t, r_t)$ *where* $q_t \sim \mathcal{N}(q_t|0, \Sigma_{f,t})$ *and* $r_t \sim \mathcal{N}(r_t|0, \Sigma_{g,t})$ *with* $\Sigma_{f,t}$ *and* $\Sigma_{g,t}$ *symmetric positive definite matrices. The choice for normally distributed germs is exemplary and other densities may be used. For many applications of practical interest the models are affine Gaussian. In this case we have* $f_t(x_t, q_t) = f_t + \mathrm{F}_{x,t}x_t + \mathrm{F}_{q,t}q_t$ *and* $g_t(x_t) = g_t + \mathrm{G}_{x,t}x_t + \mathrm{G}_{r,t}r_t$. *We assume functions* $f_t(x_t, q_t)$ *and* $g_t(x_t, r_t)$ *are one-to-one (injective) and differentiable w.r.t.* q_t *and* r_t*, respectively, for given* x_t*, and* $\dim(q_t) \leq n_x$.

2.2. Problem statement

Given a sequence of measurements $\underline{z}_T$ collected over a horizon T, our objectives are: (i) to infer the latent state trajectory $\underline{x}_T$, and (ii) to estimate the model parameters θ. The latter is typically achieved through Maximum Likelihood Estimation (MLE)

$$\theta^\star = \max_\theta \mathcal{L}_\theta(\underline{z}_T) = \max_\theta \log p_\theta(\underline{z}_T) \tag{2}$$

For probabilistic state-space models, this objective expands to the complete-data likelihood

$$\theta^\star = \max_\theta \log \int p_\theta(\underline{x}_T, \underline{z}_T) d\underline{x}_T \tag{3}$$

which is generally intractable since $\underline{x}_T$ is unobserved. To address this, we estimate the posterior distribution $p_{\hat{\theta}}(\underline{x}_T|\underline{z}_T)$ for some parameter estimate $\hat{\theta}$, and define the surrogate objective

$$\mathcal{Q}_{\theta,\hat{\theta}} = \mathbb{E}_{\hat{\theta}}\left[\log p_\theta(\underline{x}_T, \underline{z}_T)|\underline{z}_T\right] = \int \log p_\theta(\underline{x}_T, \underline{z}_T) p_{\hat{\theta}}(\underline{x}_T|\underline{z}_T) d\underline{x}_T \tag{4}$$

1. For brevity, we introduce the notations $\underline{x}_t = \{x_0, \ldots, x_t\}$ and $\overline{x}_t = \{x_t, \ldots, x_T\}$, respectively referring to a leading or trailing part of a sequence, with $t \in \mathbb{N}_0$ referring to the final or initial time instance of the corresponding subsequence. We silently assume that any complete sequence starts at time $t = 0$ and ends at time $t = T$.

Maximizing the surrogate objective (4) and updating $\hat{\theta}$ with the resulting estimate yields a sequence of parameter updates that converges to the MLE in (2); see Schön et al. (2011); Murphy (2012) for a detailed proof. The expectation in (4) is taken with respect to the smoothing density $p_{\hat{\theta}}(\underline{x}_T|\underline{z}_T)$, parameterized by the current estimate $\hat{\theta}$. Consequently, accurate state inference (objective (i)) is essential for parameter learning (objective (ii)). This iterative procedure, known as the Expectation–Maximization (EM) algorithm, is repeated until convergence, provided the required integrals and optimizations can be evaluated.

In general nonlinear and non-Gaussian settings, this expectation cannot be computed analytically and must be approximated. Particle smoothers - a class of Sequential Monte Carlo (SMC) methods - approximate these integrals using weighted trajectories Schön et al. (2011, 2015); Wills and Schön (2023). However, existing discrete-time particle smoothers exhibit several undesirable properties. First, particles typically have unequal weights, requiring complex weight updates and resampling procedures. Second, arbitrary temporal subsets do not directly represent the normalized joint posterior - while $p(x_t|\underline{z}_T)$ can be approximated, $p(x_t, x_{t'}|\underline{z}_T)$ for $t \neq t'$ cannot be obtained without additional processing. Finally, their computational cost scales quadratically with the number of particles, making them inefficient for large-scale problems Klaas et al. (2006); Cappé et al. (2007); Chopin and Papaspiliopoulos (2020).

Therefore, our goal is to efficiently approximate the smoothing density

$$p(\underline{x}_T|\underline{z}_T) \tag{5}$$

using a collection of equally weighted trajectories $\mathcal{X} = \{\underline{x}_T^n\}_{n=1}^N$ such that any temporal subset of $\mathcal{X}$ provides a consistent particle representation of the corresponding marginal or joint posterior, i.e. $\mathcal{X}_t = \{x_t^n\}_{n=1}^N \sim p(x_t|\underline{z}_T)$ and also $\mathcal{X}_t \cup \mathcal{X}_{t'} \sim \{\{x_t^n, x_{t'}^n\}\}_{n=1}^N \sim p(x_t, x_{t'}|\underline{z}_T), t \neq t'$.

2.3. Computational structures in smoothing

The smoothing densities are conventionally computed using a *forward-then-backward* recursive procedure Ho and Lee (1964); Lee (1964). This structure corresponds with how information becomes available for processing. First, the filtering densities $p(x_t|\underline{z}_t)$ are computed sequentially as new measurements arrive. These are then refined in a backward pass using the smoothing recursion

$$p(x_t|\underline{z}_T) = p(x_t|\underline{z}_t)\mathbb{E}_{p(x_{t+1}|x_t)}\left[\frac{p(x_{t+1}|\underline{z}_T)}{p(x_{t+1}|\underline{z}_t)}\right] \tag{6}$$

An alternative *backward–then–forward* structure can also be formulated. Assuming access to $p(x_0|\underline{z}_T)$, the joint posterior can be factorized as

$$p(x_t, x_{t+1}|\underline{z}_T) = p(x_t|\underline{z}_T)p(x_{t+1}|x_t, \underline{z}_T) \tag{7}$$

Marginalizing over x_t yields a forward recursion for the posterior $p(x_{t+1}|\underline{z}_T)$. This structure resembles a forward simulation governed by the *informed transition density* $p(x_{t+1}|x_t, \underline{z}_T)$, encoding information from both past and future measurements. Unlike the nominal transition $p(x_{t+1}|x_t)$, it produces trajectories that are statistically consistent with the full posterior distribution.

$$p(x_{t+1}|\underline{z}_T) = \int p(x_{t+1}|x_t, \underline{z}_T)p(x_t|\underline{z}_T)\mathrm{d}x_t \tag{8}$$

Remark 2 *Note that conditioning of the controlled transition density can be weakened to $\overline{z}_{t+1}$ due to the Markov property, i.e. the measurements, $\underline{z}_t$, cannot contain more information about the next state, x_{t+1}, than the state, x_t, itself.*

3. Related work

The most widely used nonlinear smoothing algorithms fall into two families. The Extended Kalman Smoother (EKS) Hartikainen et al. (2011) applies local linearization to extend the classical Rauch–Tung–Striebel smoother, offering computational efficiency but limited accuracy for strongly nonlinear systems. The Bootstrap Particle Smoother (BPS) Klaas et al. (2006) combines a forward particle

filter with a backward smoothing pass, handling arbitrary nonlinearities but suffering from particle degeneracy, non-uniform weights, and high computational cost Cappé et al. (2007); Chopin and Papaspiliopoulos (2020); Schön et al. (2011). The present work aims to combine the flexibility of particle-based methods with the efficiency of deterministic approaches, while producing uniformly weighted trajectories.

Our approach is rooted in dualities between Bayesian smoothing and optimal control. For hidden Itô or diffusion processes (the continuous-time analogue of a PSSM), it has long been established that the posterior can be represented by a controlled process with a modified drift term Fleming and Mitter (1982); Mitter and Newton (2003); Kim and Mehta (2023); Opper (2019), determined by solving a particular optimal control problem. A related line of work is the theory of Linearly Solvable Optimal Control (LSOC) or Path Integral Control Kappen et al. (2007); Thijssen and Kappen (2015), where the optimal policy is expressed as a conditional expectation over passively sampled trajectories. This idea has since sparked broader interest in wielding inference principles in control Theodorou and Todorov (2012); Theodorou (2015); Williams et al. (2017). A discrete-time analogue was studied independently Todorov (2007, 2008); Dvijotham and Todorov (2012) and subsequently absorbed into the broader paradigm of *probabilistic optimal control* Kárnỳ (1996); Toussaint and Storkey (2006); Rawlik et al. (2013); Rawlik (2013); Levine (2018); Watson et al. (2021); Lefebvre (2024), where probabilistic policies are obtained by projecting a joint density of the control system onto a desired density via dynamic programming. In Lefebvre (2024), the risk-seeking formulation was shown to be directly associated with smoothing Rawlik (2013); Watson et al. (2021).

The LSOC framework was used to develop a particle smoother for hidden diffusion processes Ruiz and Kappen (2017), and its use has been further explored in particle filtering Zhang et al. (2023); Taghvaei and Mehta (2023). In discrete-time, however, solving an optimal control problem to address the smoothing problem has, to the best of our knowledge, not been explored and requires different representations and techniques.

4. Smoothing with Optimal Control

In this section, we discuss an alternative approach to represent the smoothing density that leans on a connection with the theory of optimal control. The objective is to construct a synthetic control system whose closed-loop trajectories reproduce the statistical properties of the smoothing density. In this framework, the posterior distribution can be approximated by a set of equally weighted trajectories obtained through forward simulation of the control system.

To that end, first we establish how the informed transition density, $p(x_{t+1}|x_t, \overline{z}_{t+1})$ (recall Section 2.3), can be encoded as the solution of an optimization problem. This is established by projection of the posterior onto a generic representation of the synthetic control system. Second, we discuss how to parametrize the informed transition density and how to solve the associated optimization problem by leaning on a connection with probabilistic optimal control.

4.1. Open-loop transition density

First, let us define the following synthetic open-loop transition density. We will refer to it as the *controlled* transition density. Here u_{t+1} represents a synthetic control input[2]. We require of the synthetic open-loop transition density that it is sufficiently expressive.

$$p(x_{t+1}|x_t, u_{t+1}) \tag{9}$$

2. We denote the control at time t with index $t+1$ so that the sequence of controls also ranges from 0 to T. As such, the first control, u_0, can be used to encode the posterior initial density, $p(x_0|\underline{z}_T)$.

For given control, $\underline{u}_T$, this gives rise to the following joint open-loop trajectory density [3]

$$p(\underline{x}_T|\underline{u}_T) = p(x_0|u_{-1}) \prod_{t=1}^{T} p(x_t|x_{t-1}, u_t) \tag{10}$$

4.2. Distribution matching

We are now prepared to synthesize the controllers. Therefore we will *project* the joint open-loop trajectory density (10) onto the joint posterior density (5). We will accomplish this relying on the information projection Murphy (2012). This is equivalent to minimizing the relative entropy between the parameterized density and the smoothing density.

$$\underline{u}_T^\star = \arg\min_{\underline{u}_T} \mathbb{D}\left[p(\underline{x}_T|\underline{u}_T)||p(\underline{x}_T|\underline{z}_T)\right] \tag{11}$$

Remark 3 *This is a special case of the general variational inference problem where p represents the target density and q a member from a some arbitrary model density family.*

$$q^\star = \arg\min_{q\in\mathcal{Q}} \mathbb{D}[q||p]$$

Lemma 1 *Problem (11) is equivalent to the following stochastic optimal control problem*

$$\underline{u}_T^\star = \arg\min_{\underline{u}_T} \mathbb{E}\left[r_T(x_T) + \sum_{t=0}^{T-1} r_t(x_t, u_{t+1})\right]$$

starting from the initial state x_{-1} and with cost rate

$$r_t(x_t, u_{t+1}) = l(z_t|x_t) + \mathbb{D}\left[p(x_{t+1}|x_t, u_{t+1})||p(x_{t+1}|x_t)\right]$$

where $l(z_t|x_t) := -\log p(z_t|x_t)$ denotes the negative log-likelihood of the observation.

Proof Note that (11) can be simplified to

$$\underline{u}_T^\star = \arg\min_{\underline{u}_T} \mathbb{D}\left[p(\underline{x}_T|\underline{u}_T)||p(\underline{z}_T, \underline{x}_T)\right]$$

This objective can be decomposed as

$$\mathbb{D}\left[p(\underline{x}_T|\underline{u}_T)||p(\underline{z}_T, \underline{x}_T)\right] = \mathbb{E}\left[l(\underline{z}_T|\underline{x}_T)\right] + \mathbb{D}\left[p(\underline{x}_T|\underline{u}_T)||p(\underline{x}_T)\right]$$

The result immediately follows. ■

From Lemma 1 it follows that the solution of (11) is a feedback policy that we can evaluate using dynamic programming. One verifies that

$$\begin{aligned} u_{t+1}^\star(x_t) &= \arg\min_{u_t} Q_t^\star(x_t, u_{t+1}) \\ V_t^\star(x_t) &= \min_{u_{t+1}} Q_t^\star(x_t, u_{t+1}) \end{aligned} \tag{12}$$

where

$$Q_t^\star(x_t, u_{t+1}) = r_t(x_t, u_{t+1}) + \mathbb{E}\left[V_{t+1}^\star(x_{t+1})\right] \tag{13}$$

with

$$V_T(x_T) = l(z_T|x_T) \tag{14}$$

3. To simplify the presentation, we adopt a common notational convention for sequential models by omitting the root node (□). This node is a deterministic, fictitious initial state, x_{-1}, used to formally include the posterior initial state density $p(x_0|\underline{z}_T)$ within the sequential structure of the controlled transition dynamics. We refer to Appendix A.

We note that the controlled transition density $p(x_{t+1}|x_t, u_{t+1})$ is sufficiently expressive by construction when the synthetic control input u_t acts in the noise space of the original dynamics, i.e., replacing q_t in $x_{t+1} = f_t(x_t, u_t)$. Since u_t is unconstrained and occupies the same space as q_t, the controlled system can reproduce any trajectory realisable under the original stochastic model. Every feasible state transition $x_t \to x_{t+1}$ corresponds to a unique control input $\bar{u}_t(x_t, x_{t+1})$, provided $f_t(x_t, \cdot)$ is injective (see Remark 1). Since q_t and u_t live in the same space, it is implied that for every pair (x_t, x_{t+1}) there exists a corresponding u_t that would realise this transition. Consequently, the joint closed-loop density $p(\underline{x}_T|u_T^\star)$ and the joint posterior $p(\underline{x}_T|\underline{z}_T)$ are statistically equivalent by a structural consequence of the noise-space parametrization.

A detailed derivation of this recursion for the affine Gaussian PSSM, including closed-form expressions for the optimal policy, gains, and value functions, is provided in Appendix B.

4.3. Equivalence with backward-then-forward smoother

Here we demonstrate that the optimal synthetic closed-loop density is equivalent to the posterior joint density. Therefore it is sufficient to note that we have assumed that u_{t+1} completely parameterises $p(x_{t+1}|x_t, u_{t+1})$. By consequence we can also solve (11) for $p(x_{t+1}|x_t, u_{t+1})$ instead of u_{t+1}. If we can then show that the optimal controlled transition density is equivalent to the informed transition density, our statement follows.

For notational convenience we introduce $\pi_{t+1}(x_{t+1}|x_t) \equiv p(x_{t+1}|x_t, u_{t+1})$ and define

$$\underline{\pi}_T^\star = \arg\min_{\underline{\pi}_T} \mathbb{D}\left[\pi(\underline{x}_T)||p(\underline{x}_T|\underline{z}_T)\right] \tag{15}$$

where

$$\pi(\underline{x}_T) = \pi_0(x_0) \prod_{t=1}^{T} \pi_t(x_t|x_{t-1}) \tag{16}$$

As explained above, by construction we have that

$$\pi_{t+1}^\star \equiv p(x_{t+1}|x_t, u_{t+1}^\star(x_t)) \tag{17}$$

A relatively straightforward calculation then shows that indeed

$$\pi_t^\star(x_t|x_{t-1}) \equiv p(x_t|x_{t-1}, \overline{z}_{t-1}) \tag{18}$$

5. Numerical algorithm

We now develop a practical algorithm to compute a collection of trajectories $\{\underline{x}_T^n\}_{n=1}^N$ whose statistics are governed by the posterior $p(\underline{x}_T|\underline{z}_T)$. This is achieved by calculating the probabilistic state-feedback policies $\underline{\pi}_{T-1}^\star$ from equations (12)-(14) and performing N forward closed-loop simulations. By the equivalence established in Section 4.3, these trajectories form an equally weighted representation of the desired posterior representation.

The key challenge lies in computing the optimal policies. While the optimal control problem in Section 4.2 admits a closed-form solution in the affine Gaussian case (see Appendix B), it is intractable for general nonlinear systems. We therefore resort to iterative numerical solution strategies inspired by the iterative Linear Quadratic Regulator (iLQR) algorithm Li and Todorov (2004) and Differential Dynamic Programming (DDP) Jacobson and Mayne (1970).

These methods alternate between forward and backward passes to refine an affine approximation of the optimal policy. We adapt this architecture to our setting as follows:

- Forward Sweep: Execute forward closed-loop simulation using the current approximate affine Gaussian policy sequence $\underline{\pi}_T^\star$ of the form $\pi_t^\star(u_t|x_t) = \mathcal{N}(u_t|k_t^\star + \mathrm{K}_t^\star x_t, \Sigma_t^\star)$ with the actual dynamics defined in Remark (1). Given the policy uncertainty, this produces a state-action trajectory distribution $p(\underline{\xi}_T; \hat{\underline{\pi}}_{T-1}^\star)$ where $\xi_t = (x_t^\top, u_t^\top)^\top$.

- Backward Sweep: Construct a quadratic surrogate $\hat{Q}_t^\star$ by developing the expression in (13) using a second-order expansion. Once $\hat{Q}_t^\star$ is available, a quadratic approximation $\hat{V}_t^\star$ of $V_t^\star$ and affine approximation $\hat{\pi}_t^\star$ of $\pi_t^\star$ follow directly by substituting $\hat{Q}_t^\star$ into (12).

A straightforward approach builds on classical DDP and uses a second-order Taylor series expansion for developing $\hat{Q}_t^\star$ in the quadratic form of (22). We construct this approximation around the sample mean, $\hat{\xi}_t$, of a set of sampled state-action trajectories $\{\underline{\xi}_T^n\}_{n=1}^N \sim p(\underline{\xi}_T; \hat{\underline{\pi}}_{T-1}^\star)$, drawn from a nominal policy $\hat{\underline{\pi}}_{T-1}^\star$. The resulting backward recursion is obtained by solving (12) for the nonlinear dynamics, yielding the following coefficients in the quadratic approximation

$$
\begin{aligned}
Q_{0,t} &= -\log|\Sigma_{uu,t+1}| + \operatorname{tr}\left(Q_{uu,t}\Sigma_{uu,t+1}\right) + \text{cst.} \\
Q_{x,t} &= \frac{1}{2}\mathrm{G}_{x,t}^\top(\mathrm{G}_{r,t}^{-\top}\Sigma_{g,t}^{-1}\mathrm{G}_{r,t}^{-1} + \mathrm{G}_{r,t}^{-\top}\Sigma_{g,t}^{-T}\mathrm{G}_{r,t}^{-1})(g_t(\hat{x}_t) - z_t) + \tilde{\mathrm{F}}_{x,t}^\top V_{x,t+1} \\
&\quad + \frac{1}{2}\tilde{\mathrm{F}}_{x,t}^\top(V_{xx,t+1} + V_{xx,t+1}^T)(f_t(\hat{x}_t, \hat{u}_t) - \hat{x}_{t+1}) \\
Q_{u,t} &= \Sigma_{f,t}^{-1}u_t + \tilde{\mathrm{F}}_{u,t}^\top V_{x,t+1} + \frac{1}{2}\tilde{\mathrm{F}}_{u,t}^\top(V_{xx,t+1} + V_{xx,t+1}^T)(f_t(\hat{x}_t, \hat{u}_t) - \hat{x}_{t+1}) \\
Q_{xx,t} &= \mathrm{G}_{x,t}^\top\mathrm{G}_{r,t}^{-\top}\Sigma_{g,t}^{-1}\mathrm{G}_{r,t}^{-1}\mathrm{G}_{x,t} + \tilde{\mathrm{F}}_{x,t}^\top V_{xx,t+1}\tilde{\mathrm{F}}_{x,t} \\
Q_{ux,t} &= \tilde{\mathrm{F}}_{u,t}^\top V_{xx,t+1}\tilde{\mathrm{F}}_{x,t} \\
Q_{uu,t} &= \Sigma_{f,t}^{-1} + \tilde{\mathrm{F}}_{u,t}^\top V_{xx,t+1}\tilde{\mathrm{F}}_{u,t}
\end{aligned}
\tag{19}
$$

where $\tilde{F}_t = [\tilde{F}_{x,t} \mid \tilde{F}_{u,t}]$ denotes the partition of the ensemble gradient into state and control components, with $\tilde{F}_{x,t} \in \mathbb{R}^{n_x \times n_x}$ and $\tilde{F}_{u,t} \in \mathbb{R}^{n_x \times n_u}$.

To evaluate these terms, we require a linear approximation of the nonlinear dynamics. This is achieved using the trajectory ensemble to construct a sample-based linear model of the form $x_{t+1} = \tilde{\mathrm{F}}_t\xi_t$ where

$$
\tilde{\mathrm{F}}_t = \Delta\mathrm{X}_{t+1}(\Delta\Xi_t^\top\Delta\Xi_t)^{-1}\Delta\Xi_t^\top \tag{20}
$$

with $\Delta\Xi_t = (\Delta\xi_t^1, \ldots, \Delta\xi_t^N)^\top$ and $\Delta\xi_t = \xi_t - \hat{\xi}_t$. We refer to $\tilde{F}_t$ as the *ensemble gradient*.

In the forward sweep, the system is simulated forward in time and the optimal affine feedback control policy $\pi_t^\star(u_t|x_t)$ is then determined by (24). Substituting this policy into (12) then yields the backward recursive expressions described by (25). This method is summarized in Algorithm 1.

The key advantage of the EG-PDPS over the Bootstrap Particle Smoother (BPS) Klaas et al. (2006) is its linear computational scaling with the number of particles N. Specifically, EG-PDPS exhibits complexity $O(NTn_\xi^2)$, where $n_\xi = \dim(\xi)$, compared to the quadratic $O(N^2T)$ scaling of BPS. For typical cases where $N \gg n_\xi$, this yields substantial computational savings.

6. Results

We evaluate the proposed methods for computing smoothing densities and demonstrate their use in system identification and model learning. Experiments are performed on three simulated systems: a linear Gaussian model and two nonlinear systems, and on an experimental robotic drivetrain. The system descriptions, model parameters, and initialization details are provided in Appendix C. We assess accuracy, computational efficiency, and performance in an EM–based identification task. All implementations use PyTorch and run on a 3.2 GHz Intel Core i9-14900K CPU.

Algorithm 1 EG-PDPS

Require: PSSM, $\mathcal{M}$; measurements, $\underline{z}_T$; initial policies, $\hat{\underline{\pi}}^\star_{T-1}$ (e.g. $\underline{\rho}_{T-1}$); particle count N
Ensure: $\{\underline{x}_T\} \sim p(\underline{x}_T|\underline{z}_T)$
while !converged **do**
  Forward sweep
  for $n \leftarrow 1$ **to** N **do**
    Sample $x_0^n \sim \hat{\pi}^\star_{-1}(x_0|\square)$
    for $t \leftarrow 0$ **to** $T-1$ **do**
      Sample $u_t^n \sim \hat{\pi}^\star_t(u_t|x_t^n)$
      Propagate $x_{t+1}^n \leftarrow f_t(x_t^n, u_t^n)$
    end for
  end for
  Compute means $\hat{\xi}_t$ and deviations $\Delta\Xi_t$ from $\{\xi_t^n\}_{n=1}^N$
  Compute ensemble gradients $\tilde{F}_t$ (20)
  Backward sweep
  Evaluate $\hat{V}_T^\star$ from terminal cost (14)
  for $t \leftarrow T-1$ **to** 0 **do**
    Evaluate coefficients of $\hat{Q}_t^\star$ using $\tilde{F}_t$ and $\hat{V}_{t+1}^\star$ (19)
    Evaluate $\hat{V}_t^\star$ from $\hat{Q}_t^\star$ (25)
    Compute control gains and update policy $\hat{\pi}_t^\star(u_t|x_t) = \mathcal{N}(u_t|k_t^\star + K_t^\star x_t, \Sigma_t^\star)$ (24)
  end for
end while

6.1. Posterior Density Estimation

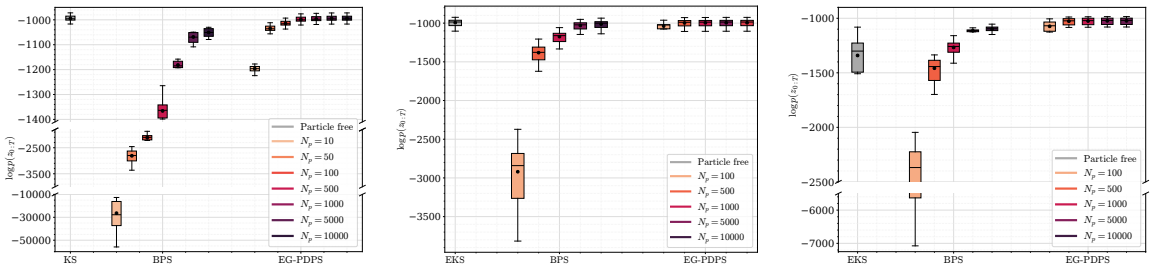
6.1.1. ESTIMATION ACCURACY

The performance of the proposed algorithm is evaluated using the log-likelihood $\log p(\underline{z}_T)$ and compared with the (Extended) Kalman Smoother (EKS) Hartikainen et al. (2011) and the Bootstrap Particle Smoother (BPS) Klaas et al. (2006). For fair comparison, the state trajectories $\underline{x}_T$ are initialized from a particle filter[4]. Each system is simulated over ten trajectories of length $T = 1000$, initialized to sufficiently excite dynamic behavior.

Fig. 1 shows the log-likelihood across the different smoothing algorithms.For the linear Gaussian double MSD system, the Kalman Smoother provides the exact posterior, serving as an upper performance bound. One can verify that for linear Gaussian systems, the smoothing distributions obtained by running Algorithm 1 with the exact system equations instead of the sample-based approximations, results in equivalent distributions to those obtained by the KS. The proposed EG-PDPS algorithm closely approaches this bound with a modest number of particles, demonstrating its accuracy and superior sample efficiency relative to BPS.

For the nonlinear Lorenz system, EG-PDPS achieves performance comparable to (with as few as 500 particles) or better (with 10,000 particles) than the EKS, whereas BPS requires an order of magnitude more particles and still underperforms relative to both EKS and EG-PDPS. In the highly nonlinear double pendulum case, EG-PDPS again demonstrates substantially higher accuracy - outperforming BPS even when using 100 particles versus 10,000 for BPS - while the EKS struggles due to its linearization-based approximation.

4. This initialization starts the algorithm with the backward sweep.



(a) Double MSD (b) Lorentz system (c) Double pendulum

Figure 1: Log-likelihood comparison of the posterior density across multiple smoothing algorithms.

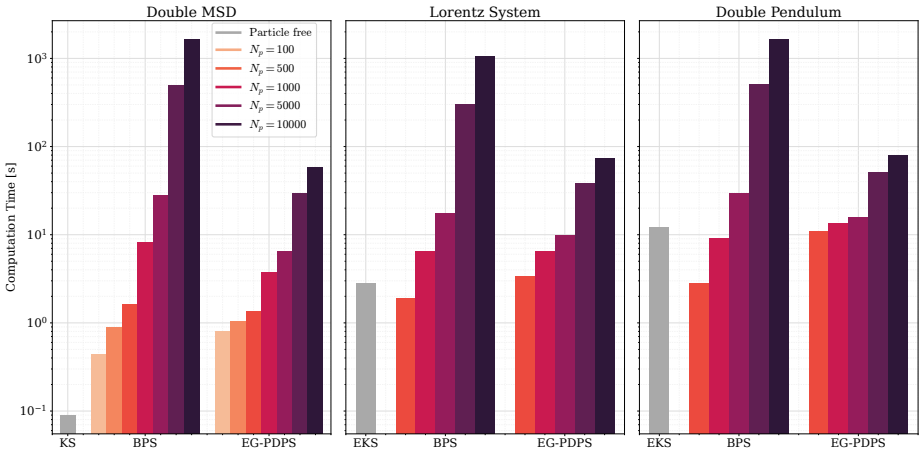


Figure 2: Average run times until convergence for different smoothing algorithms.

6.1.2. Computational demand

The computational efficiency of the proposed methods is assessed in terms of average runtime. Fig. 2 shows the computation time (in s) required for convergence of each smoothing algorithm on a single 1000-step trajectory, averaged over 10 runs. The BPS exhibits quadratic complexity with respect to the number of particles, causing computation time to grow rapidly. In contrast, EG-PDPS scales linearly with particle count, resulting in significantly lower computational demand.

6.2. System identification

This section demonstrates the utility of the proposed particle-based smoothing approach for probabilistic system identification. Specifically, we estimate the parameters of nonlinear dynamical systems through Maximum Likelihood optimization within an Expectation–Maximization framework, inspired by the formulation in Schön et al. (2011).

In the EM algorithm, the Expectation step requires computing the minimum-variance estimate of the joint log-likelihood $\log p(\underline{x}_T, \underline{z}_T)$, or Evidence Lower Bound (ELBO), as described in Section 2.2. This expectation depends on the smoothing density $p_{\hat{\theta}}(\underline{x}_T|\underline{z}_T)$, which we approximate using the EG-PDPS algorithm. Compared to the implementation detailed in Schön et al. (2011) based on the BPS, the proposed method produces uniformly weighted trajectories, eliminating the need for importance weighting or resampling. Combined with the linear computational complexity of EG-PDPS, this results in a highly efficient and numerically stable EM procedure. Implementation details are provided in Appendix D.

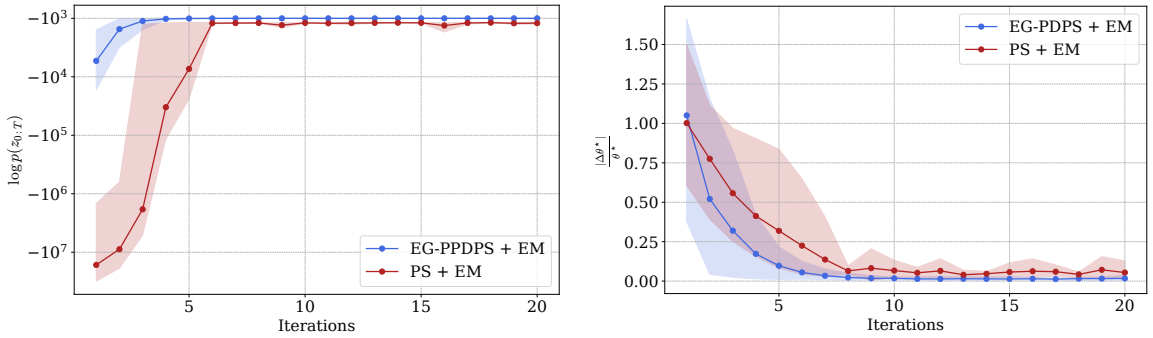
We evaluate the method on the Lorenz system, governed by

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \sigma(x_2 - x_1) \\ x_1(\rho - x_3) - x_2 \\ x_1 x_2 - \beta x_3 \end{bmatrix}, \quad w \sim \mathcal{N}(\mathrm{O}_3, q\mathrm{I}_3), \tag{21}$$

with true parameters $\theta^\star = [\alpha^\star, \beta^\star, \rho^\star, q^\star] = [10, \frac{8}{3}, 28, 0.1]$. The EM algorithm (described in Algorithm 2 in Appendix D) is executed ten times with randomized initial parameters sampled from $\mathcal{N}(\theta^\star, 0.5^2\theta^\star)$, using $N_p = 1000$ particles per run.

Fig. 3 compares the convergence of the standard BPS-based EM algorithm with our EG-PDPS-based EM implementation. The improved posterior estimation in the E-step provides smoother gradients and a more stable optimization landscape, leading to faster and more consistent convergence of the log-likelihood and a higher final value. The parameter mean-squared error decreases monotonically for both methods, but EG-PDPS achieves faster convergence with lower variance across trials, reflecting improved reliability and accuracy.

Since EG-PDPS generates complete, uniformly weighted trajectories, the transition likelihood can be evaluated independently along each trajectory. This reduces the computational complexity of evaluating $\mathcal{Q}_{\theta,\hat{\theta}}$ to $O(NT)$ (see Appendix D), compared to $O(N^3T)$ for the Bootstrap Particle Smoother, which requires pairwise importance weighting Schön et al. (2011). Beyond faster computation, the EG-PDPS–based EM algorithm provides smoother, noise-free parameter updates when batch-optimizing the ELBO and converges significantly faster. In our experiments, EG-PDPS+EM converged on average in nine iterations (112.8 s per iteration, total 1015.2 s) versus thirteen iterations for BPS+EM (2757.6 s per iteration, total 35,848.8 s).



(a) Likelihood

(b) MSE

Figure 3: Parameter identification results for the Lorentz system using the EM algorithm.

7. Conclusion

We introduced an efficient sample-based smoothing framework that reformulates Bayesian state estimation as a probabilistic optimal control problem. The resulting algorithm computes an optimal stochastic policy through probabilistic dynamic programming, enabling forward simulation of equally weighted trajectories that represent the posterior distribution without backward passes or importance weighting. The method scales linearly with the number of particles and provides a uniformly weighted posterior representation, simplifying and accelerating Expectation–Maximization–based system identification. Experiments on nonlinear systems confirm accurate state and parameter estimation at substantially reduced computational cost.

Acknowledgments

This work was supported by the Flanders AI Research Program, the Research Foundation Flanders (FWO) project CTRLxAI (SBO grant no. S007723N), and the CoMoDO IRVA project of Flanders Make, the strategic research centre for the manufacturing industry of Flanders, Belgium.

References

O. Cappé, S. J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

N. Chopin and O. Papaspiliopoulos. *An introduction to sequential Monte Carlo*, volume 4. Springer, 2020.

K Dvijotham and E Todorov. Linearly solvable optimal control. In *Reinforcement learning and approximate dynamic programming for feedback control*, volume 17, pages 119–141. Citeseer, 2012.

W. H. Fleming and S. K. Mitter. Optimal control and nonlinear filtering for nondegenerate diffusion processes. *Stochastics: An International Journal of Probability and Stochastic Processes*, 8(1): 63–77, 1982.

J. Hartikainen, A. Solin, and S. Särkkä. *Optimal filtering with Kalman filters and smoothers–a Manual for Matlab toolbox EKF/UKF*. 2011.

Y. C. Ho and R. Lee. A Bayesian approach to problems in stochastic estimation and control. *IEEE transactions on automatic control*, 9(4):333–339, 1964.

D. H. Jacobson and D. Q. Mayne. Differential dynamic programming. 1970.

H. Kappen, W. Wiegerinck, and B. van den Broek. A path integral approach to agent planning. *Autonomous Agents and Multi-Agent Systems*, 2007.

J. W. Kim and P. G. Mehta. Duality for nonlinear filtering II: Optimal control. *IEEE Transactions on Automatic Control*, 69(2):712–725, 2023.

M. Klaas, M. Briers, N. De Freitas, A. Doucet, S. Maskell, and D. Lang. Fast particle smoothing: If I had a million particles. In *International conference on Machine learning*, pages 481–488, 2006.

M. Kárnỳ. Towards fully probabilistic control design. *Automatica*, 32(12):1719–1722, 1996.

R. C. K. Lee. *Optimal estimation, identification, and control*, volume 1964. MIT press Cambridge, MA, 1964.

T. Lefebvre. Probabilistic control and majorisation of optimal control. *Systems & Control Letters*, 190:105837, 2024.

S. Levine. Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review. *arXiv preprint, arXiv:1805.00909*, 2018.

W. Li and E. Todorov. Iterative linear quadratic regulator design for nonlinear biological movement systems. In *First International Conference on Informatics in Control, Automation and Robotics*, volume 2, pages 222–229. SciTePress, 2004.

E. Lorentz. Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, 20:130–141, 1963.

S. K. Mitter and N. J. Newton. A variational approach to nonlinear estimation. *Journal on control and optimization*, 42(5):1813–1833, 2003.

K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Manfred Opper. Variational inference for stochastic differential equations. *Annalen der Physik*, 531 (3):1800233, 2019.

K. Rawlik, M. Toussaint, and S. Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference (extended abstract). In *International Joint Conference on Artificial Intelligence*, IJCAI 2013, pages 3052–3056, 2013.

K. C. Rawlik. On probabilistic inference approaches to stochastic optimal control. 2013. The University of Edinburgh.

Conor Rosato, Lee Devlin, Vincent Beraud, Paul Horridge, Thomas B Schön, and Simon Maskell. Efficient learning of the parameters of non-linear models using differentiable resampling in particle filters. *IEEE Transactions on Signal Processing*, 70:3676–3692, 2022.

H. C. Ruiz and H. J. Kappen. Particle smoothing for hidden diffusion processes: Adaptive path integral smoother. *IEEE Transactions on Signal Processing*, 65(12):3191–3203, 2017.

T. B. Schön, A. Wills, and B. Ninness. System identification of nonlinear state-space models. *Automatica*, 47(1):39–49, 2011.

T. B. Schön, F. Lindsten, J. Dahlin, J. Wågberg, C. A. Naesseth, A. Svensson, and L. Dai. Sequential Monte Carlo methods for system identification. *IFAC-PapersOnLine*, 48(28):775–786, 2015.

Thomas B Schön, Andreas Svensson, Lawrence Murray, and Fredrik Lindsten. Probabilistic learning of nonlinear dynamical systems using sequential Monte Carlo. *Mechanical systems and signal processing*, 104:866–883, 2018.

M.W. Spong and M. Vidyasagar. Robot dynamics and control. *Wiley & Sons, Inc*, 1989.

S. Särkkä. *Bayesian filtering and smoothing*. Number 3. Cambridge University Press, 2013.

Amirhossein Taghvaei and Prashant G Mehta. A survey of feedback particle filter and related controlled interacting particle systems (CIPS). *Annual Reviews in Control*, 55:356–378, 2023.

E. A. Theodorou. Nonlinear stochastic control and information theoretic dualities: Connections, interdependencies and thermodynamic interpretations. *Entropy*, 17(5):3352–3375, 2015.

E. A. Theodorou and E. Todorov. Relative entropy and free energy dualities: Connections to path integral and kl control. In *IEEE Conference on decision and control (cdc)*, pages 1466–1473. IEEE, 2012.

S. Thijssen and H. J. Kappen. Path integral control and state-dependent feedback. *Physical Review E*, 91(3):032104, 2015.

E. Todorov. Linearly-solvable Markov decision problems. In *Advances in neural information processing systems*, pages 1369–1376, 2007.

E. Todorov. General duality between optimal control and estimation. In *IEEE Conference on Decision and Control*, pages 4286–4292, 2008.

M. Toussaint and A. Storkey. Probabilistic inference for solving discrete and continuous state Markov Decision Processes. In *International Conference of Machine Learning (ICML)*, 2006.

V. Vantilborgh, T. Lefebvre, Kerem E., and G. Crevecoeur. Data-driven virtual sensing for probabilistic condition monitoring of solenoid valves. *IEEE Transactions on Automation Science and Engineering*, 21(2):1297–1311, 2024.

Joe Watson, Hany Abdulsamad, Rolf Findeisen, and Jan Peters. Efficient stochastic optimal control through approximate Bayesian input inference. *arXiv preprint arXiv:2105.07693*, 2021.

Grady Williams, Nolan Wagener, Brian Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou. Information theoretic mpc for model-based reinforcement learning. In *IEEE international conference on robotics and automation (ICRA)*, pages 1714–1721. IEEE, 2017.

A. G. Wills and T. B. Schön. Sequential monte carlo: A unified review. *Annual Review of Control, Robotics, and Autonomous Systems*, 6(1):159–182, 2023.

Q. Zhang, A. Taghvaei, and Y. Chen. An optimal control approach to particle filtering. *Automatica*, 151:110894, 2023.

Appendix A. The Root Node and Initial Density

In the probabilistic optimal control formulation Section 4, the joint open-loop trajectory density $p(\underline{x}_T|\square, \underline{u}_T)$ is defined over the sequence $\underline{x}_T$. Formally, this sequence begins with the distribution $p(x_0|\square, u_0)$, which is parameterized by the first synthetic control input, u_0.The explicit inclusion of $\square$ allows the posterior initial density to be defined as $p(x_0|\underline{z}_T) \equiv p(x_0|\square, u_0^\star)$ where $u_0^\star$ is the optimal synthetic control input at time $t = 0$. This ensures that the dynamic programming recursion (12)–(14) holds consistently for all time steps from $t = 0$ to T.

In the main text, we follow the convention of absorbing the root node into the initial state x_0. This results in the following notational equivalence for the key densities:

Table 1: Notational equivalence when absorbing the root node into the initial state x_0.

Formal Notation (with $\square$)	**Simplified Notation (Main Text)**	**Description**
$p(\underline{x}_T, \square, \underline{u}_T)$	$p(\underline{x}_T, \underline{u}_T)$	Joint state–control density
$p(\underline{x}_T, \square, \underline{z}_T)$	$p(\underline{x}_T, \underline{z}_T)$	Joint state–measurement density
$x_{-1} = \square$	Omitted	Fictitious initial state

This simplification allows us to present the results of Section 4, such as the stochastic optimal control problem in (11), starting directly from $t = 0$, implicitly assuming that x_0 and the corresponding first control u_0 encode the necessary posterior information. The dynamic programming steps in Equations (12)–(14) are understood to apply starting at $t = 0$, with the terminal condition $V_T(x_T)$ defined at time T.

Appendix B. Specialization to affine Gaussian PSSMs

In this section we can specialize the result in Lemma 1 to the affine Gaussian PSSM that was defined in remark 1. In this setting we have that the optimal policy is affine Gaussian, that is $\pi_t^\star(u_t|x_t) = \mathcal{N}(u_t|k_t^\star + \mathrm{K}_t^\star x_t, \Sigma_t^\star)$. The functions, $V_t^\star$ and $Q_t^\star$, are quadratic in their arguments as follows

$$Q_t^\star(\xi_t) = \frac{1}{2}\begin{pmatrix}1\\ \xi_t\end{pmatrix}^\top \begin{pmatrix}Q_{0,t} & Q_{\xi,t}^\top \\ Q_{\xi,t} & Q_{\xi\xi,t}\end{pmatrix}\begin{pmatrix}1\\ \xi_t\end{pmatrix} \tag{22a}$$

$$V_t^\star(x_t) = \frac{1}{2}\begin{pmatrix}1\\ x_t\end{pmatrix}^\top \begin{pmatrix}V_{0,t} & V_{x,t}^\top \\ V_{x,t} & V_{xx,t}\end{pmatrix}\begin{pmatrix}1\\ x_t\end{pmatrix} \tag{22b}$$

in which $\xi_t = (x_t^\top, u_t^\top)^\top$ is the state-control pair. Further, the following matrix partitions are implied.

$$Q^\star_{\xi,t} = \begin{pmatrix} Q_{x,t} \\ Q_{u,t} \end{pmatrix} \tag{23a}$$

$$Q^\star_{\xi\xi,t} = \begin{pmatrix} Q_{xx,t} & Q_{ux,t}^\top \\ Q_{ux,t} & Q_{uu,t} \end{pmatrix} \tag{23b}$$

With the prior policy defined as $\rho_t^\star(u_t|x_t) = \mathcal{N}(u_t|0, \Sigma_{f,t})$ and using (12), it follows that

$$k_t^\star = -\Sigma_t^\star Q_{u,t} \tag{24a}$$

$$\mathrm{K}_t^\star = -\Sigma_t^\star Q_{ux,t} \tag{24b}$$

$$\Sigma_t^\star = Q_{uu,t}^{-1} \tag{24c}$$

where

$$V_{x,t} = Q_{x,t} - \mathrm{K}_t^{\star\top} \Sigma_t^{\star,-1} k_t^\star \tag{25a}$$

$$V_{xx,t} = Q_{xx,t} - \mathrm{K}_t^{\star\top} \Sigma_t^{\star,-1} \mathrm{K}_t^\star \tag{25b}$$

and using (13), we have

$$Q_{\xi,t} = C_{\xi,t} + \mathrm{F}_t^\top \left(V_{x,t+1} + V_{xx,t+1} f_t\right) \tag{26a}$$

$$Q_{\xi\xi,t} = C_{\xi\xi,t} + \mathrm{F}_t^\top V_{xx,t+1} \mathrm{F}_t \tag{26b}$$

where $\mathrm{F}_t = (\mathrm{F}_{x,t}^\top, \mathrm{F}_{q,t}^\top)^\top$, $C_{\xi,t}$ and $C_{\xi\xi,t}$ are the cost function coefficients, which correspond to the first and second-order terms for ξ_t. These coefficients are partitioned as shown in (23a), with the corresponding partitioned terms for the control input being zero. For the affine Gaussian PSSM defined in remark 1, the cost function is given by

$$c_t(x_t) = \frac{1}{2}(z_t - \mathrm{G}_{x,t} x_t - g_t)^\top \mathrm{G}_{r,t}^{-\top} \Sigma_{g,t}^{-1} \mathrm{G}_{r,t}^{-1} (z_t - \mathrm{G}_{x,t} x_t - g_t). \tag{27}$$

Remark 4 *In the affine Gaussian setting, the posterior initial state density, $p(x_0|\underline{z}_T)$, is equivalent to $\mathcal{N}(k_{-1}^\star, \Sigma_{-1}^\star)$.*

Remark 5 *The equations for solving the affine Gaussian problem can be extended to approximately solve nonlinear systems by performing iterated linearization as in e.g. Li and Todorov (2004).*

Appendix C. System Descriptions

C.1. Double Mass-Spring-Damper (MSD)

A linear four-state, two-output system, with dynamics and model parameters as described in Vantilborgh et al. (2024)[5], discretized with sample time $T_s = 0.01$s. The initial state is sampled from a distribution, $\mathcal{N}((1 \quad 1 \quad 1), 10^{-2}\mathrm{I}_3)$. This is profiled since an exact solution of the posterior density can be computed in the linear Gaussian case.

C.2. Lorentz systems

A three-state nonlinear with dynamics and model parameters defined according to the classical Lorentz system Lorentz (1963), a partial observation function defined by a randomly generated 2 by 3 matrix and $T_s = 0.01$s. The initial state follows $\mathcal{N}((10 \quad 10 \quad -10), \mathrm{I}_3)$.

5. In this work we use a partial observation function, where only the position and velocity of the first mass are observed.

C.3. Double Pendulum

A highly nonlinear four-state system Spong and Vidyasagar (1989) with lengths $= 1m$ and masses $= 2kg$, integrated at $T_s = 0.02$s. Only the angular positions θ_1 and θ_2 of the first and second pendulums are partially observed, with the observation model defined as $p(z_t|x_t) = \mathcal{N}((\sin(\theta_1) \quad \sin(\theta_2)), 10^{-2}\mathrm{I}_2)$. The initial state is sampled from a distribution $\mathcal{N}((\pi \quad 0 \quad 0 \quad 0), 0.5\mathrm{I}_4)$.

Appendix D. Expectation-Maximization

Here we provide the implementation details for obtaining the Maximum Likelihood estimate of the model parameters θ using the Expectation-Maximization algorithm, based on the developments in this work.

In the E-step, the surrogate objective $\mathcal{Q}_{\theta,\hat{\theta}}$ is evaluated. Substituting the joint distribution (1) into (4), the objective decomposes as

$$\mathcal{Q}_{\theta,\hat{\theta}} = \mathcal{Q}_0 + \mathcal{Q}_x + \mathcal{Q}_z \tag{28}$$

Posterior densities $p_{\hat{\theta}}(\underline{x}_T|\underline{z}_T)$ are computed using Algorithm 1. As established in 4.3, the resulting set of equally weighted trajectories $\mathcal{X} = \{\underline{x}_T^n\}_{n=1}^N$ allows straightforward Monte Carlo approximation of each term

$$\mathcal{Q}_0 \approx \frac{1}{N}\sum_{n=1}^{N} \log p_\theta(x_0^n) \tag{29a}$$

$$\mathcal{Q}_x \approx \frac{1}{N}\sum_{t=1}^{T}\sum_{n=1}^{N} \log p_\theta(x_{t+1}^n|x_t^n) \tag{29b}$$

$$\mathcal{Q}_z \approx \frac{1}{N}\sum_{t=0}^{T}\sum_{n=1}^{N} \log p_\theta(z_t|x_t^n) \tag{29c}$$

Note that since EG-PDPS produces complete trajectory samples with uniform weights, we evaluate the transition likelihood along each trajectory independently. This yields a computational complexity of $O(NT)$ for evaluating $\mathcal{Q}_{\theta,\hat{\theta}}$, compared to the $O(N^3T)$ complexity of the Bootstrap Particle Smoother, which requires importance weighting between all particle pairs Schön et al. (2011).

Subsequently, the M-step involves maximizing $\mathcal{Q}_{\theta,\hat{\theta}}$ with respect to the parameters θ. In general, finding a closed form solution for this maximization is infeasible beyond the linear Gaussian case. To address this, we resort to the optimization tools available in the PyTorch library to efficiently optimize the differentiable system models.

The developments outlined above are summarized in Algorithm 2, which presents the EM algorithm for particle-based probabilistic system identification in nonlinear systems.

Algorithm 2 Expectation-Maximization for parameter identification

Require: Initial $\hat{\theta}$
1: **while** !converged **do**
2: E-step:
compute $p_{\hat{\theta}}(x_t|\underline{z}_T)$ using Algorithm 1
compute $Q(\theta, \hat{\theta})$ using (29)
3: M-step:
$\theta^\star = \arg\max_\theta Q(\theta, \hat{\theta})$
4: $\hat{\theta} \leftarrow \theta^\star$
5: **end while**