Proceedings of Machine Learning Research vol xxx:1–12, 2021

Particle MPC for Uncertain and Learning-Based Control

Robert Dyro RDYRO@STANFORD.EDU
James Harrison JHARRISON@STANFORD.EDU
Apoorva Sharma APOORVA@STANFORD.EDU
Marco Pavone PAVONE@STANFORD.EDU
Stanford University, Stanford, CA

Abstract

Autonomous decision-making in novel or changing environments requires quantification and consideration of uncertainties in the system or environment that impact downstream control performance. Thus, as robotic systems move from highly structured environments to open worlds, incorporating uncertainty in learning or estimation into the control pipeline is essential for robust and efficient performance. In this paper we present a nonlinear particle model predictive control (PMPC) approach to control under uncertainty. This approach, due to the particle representation of uncertainty, is capable of handling arbitrary uncertainty specifications. We implement our nonlinear PMPC scheme with a sequential convex programming non-convex optimization scheme, and we discuss practical implementation of such a framework. We investigate our approach for two robotic systems across three problem settings: time-varying, partially observed dynamics; sensing uncertainty; and model-based reinforcement learning, and show that our approach improves performance over baselines in all settings.

Keywords: Control, Decision-Making under Uncertainty, Reinforcement Learning, Model Predictive Control

1. Introduction

As autonomous decision-making agents such as robots leave narrowly tailored environments and begin operating in unstructured, uncertain worlds, incorporation of uncertainty into the decision-making pipeline will be essential to intelligently trade off risk and reward. Uncertainty is ubiquitous in robotic systems; from state estimation, to fault detection, to dynamics learning, approximate Bayesian estimation and filtering methods hold central roles throughout the autonomy stack. Moreover, as robots regularly enter novel environments, interact with humans, or perform novel tasks, uncertainty and ambiguity will be omnipresent during operation.

Incorporating uncertainty into the decision-making pipeline is necessary to ensure systems are accurately considering the full distribution of possible outcomes when choosing actions. Despite the benefits of this probabilistic reasoning, *certainty equivalent* methods—in which a point estimate is used in downstream control—still dominate. Outside of these methods, *robust* methods, which consider adversarial disturbances at each time step, are common. However, these methods are typically over-conservative due to their worst-case, set theoretic approach. In this work, we investigate particle-based model predictive control that enables tunable conservatism and pairs naturally with Bayesian models common in state estimation and dynamics learning such as sequential (particle filter) Monte Carlo and ensemble neural networks. Our approach is based on a discrete particle representation of distributions and is thus applicable to arbitrary representations of uncertainty.

1.1. Contributions

In this work we combine sequential convex programming (SCP) with a particle model predictive control (PMPC) formulation to efficiently solve nonlinear control problems with arbitrary uncertainty in state measurement or dynamics models. We argue that this formulation, based on Monte Carlo distributional approximations as opposed to methods based on set theoretic uncertainty or approximation with analytically tractable distributions, is a highly flexible and effective approach to control under uncertainty. Our approach relies on a variable *consensus horizon*: the period over which the control actions for each particle must agree, which allows a tunable parameter governing the degree of conservatism and implicitly accounting for the effects of future information gain. We investigate the role of the the consensus horizon both theoretically—showing that there exists MDPs for which only a minimally long consensus horizon yields optimal behavior—as well as experimentally.

We investigate three problem settings: learning-based control (or equivalently, model-based reinforcement learning) in which an agents learns a model of the system dynamics from nothing; time-varying parametric uncertainty such as faults or wind gusts; and sensing uncertainties such as those arising from particle filter state estimation. All three problems are investigated on two different systems, and we find substantial performance gains from our approach relative to standard certainty equivalent and uncertainty-aware sampling-based control schemes.

2. Problem Formulation

In this work we will consider control of partially observed dynamical systems, where the partially observed component may be stationary (as in dynamics learning for a black box system) or dynamic (as in fault detection or state estimation). We denote the system state at time step j as $x^{(j)} \in \mathbb{R}^s$, the action taken from this state $u^{(j)} \in \mathbb{R}^a$, and the observation as $o^{(j)} \in \mathbb{R}^o$. We will assume throughout this work that only the observations are observed.

The system has nonlinear, discrete time dynamics

$$x^{(j+1)} = f(x^{(j)}, u^{(j)}, w^{(j)}), \qquad o^{(j+1)} = g(x^{(j)}, v^{(j)})$$

where $f(\cdot)$ and $g(\cdot)$ are the state dynamics and observation function respectively, and $w^{(j)}, v^{(j)}$ are stochastic disturbances. Note that this is a very general POMDP formulation—we will describe more specific dynamics or measurement structures in our experimental evaluation.

We encode our control task in the form of a stage-wise cost function $c^{(j)}(x^{(j)}, u^{(j)})$ and state and action constraints $x^{(j)} \in \mathbb{X}^{(j)}$, $u^{(j)} \in \mathbb{U}^{(j)}$. This gives rise to the finite horizon optimal control problem

$$\begin{aligned} \text{minimize} \quad & \sum_{j=0}^{N} \mathbb{E}_{x^{(j)}} \left[c^{(j)}(x^{(j)}, u^{(j)}) \right] \\ \text{subject to} \quad & x^{(j+1)} = f(x^{(j)}, u^{(j)}) \\ & p(x^{(j)} \in \mathbb{X}^{(j)} \ \forall j) \geq 1 - \alpha \\ & u^{(j)} \in \mathbb{U}^{(j)} \ \forall j \end{aligned} \tag{1}$$

where $\alpha \in [0, 1]$ governs the degree of probabilistic constraint satisfaction.

As many robotics systems rely on Bayes filters to maintain uncertainty estimates in dynamics or state, we assume that $x^{(0)} \sim \mathcal{X}_0$, where $\mathcal{X}_0$ are arbitrary distributions over state and dynamics. Note we assume no distributional form, as complex multi-modal distributions are commonplace in robotics.

3. Approach

The fundamental challenges in our problem statement arise from (1) the presence of uncertainty (over possibly time-varying unknown state elements) and (2) the nonlinear dynamics. In this section, we detail our algorithmic approach, first focusing on how we handle uncertainty, and then discussing how we deal with the nonlinear dynamics.

3.1. Handling Uncertainty through Particle Model Predictive Control

In designing a control strategy for this problem, we need an approach which can factor in the temporally correlated effects of state uncertainty. At the same time we need an approach which remains computationally efficient, so that we can replan online to handle deviations due to the inevitable model mismatch and/or factor in online updates in state uncertainty from online estimation. In order to do so, we propose to take a particle-based approach, wherein uncertainty representations of initial state and dynamics are fed into the control algorithm as a collection of particles.

Formally, our particle model predictive control (PMPC) approach has the structure of the finite horizon optimal control problem (1). We approximate the uncertainty in the initial state by a collection of particles. Each particle represents the evolution of the system under one realization of the initial state. That is, for particle i, the state evolves according to $x_i^{(j+1)} = f(x_i^{(j)}, u_i^{(j)})$, where $x_i^{(0)} \sim \mathcal{X}_0$. By propagating each particle separately in time, we account for the time correlated effects of dynamics and state uncertainty in contrast to approaches which handle uncertainty through uncorrelated disturbances at each time step. Importantly, for each particle we get a *separate trajectory* with potentially different states and actions.

Given these multiple objectives, we approximate the expectation in (1) via a (potentially weighted) sum of cost across these particles. For small values of the constraint satisfaction probability α_x, the problem becomes increasingly close to a purely set-theoretic problem in which the image of the initial set (summed with the support of the disturbance) under the dynamics must satisfy the constraints. This leads to a central tension as many inferential frameworks rely on infinite support distributions, while this renders constraint satisfaction impossible. We sidestep rigorous satisfaction of set-theoretic constraints in our framework; instead, we enforce constraint satisfaction for all particles. In our experiments, we found that this approach was sufficient, but developing practical guarantees on probabilistic constraint satisfaction.

Even as we simulate several possible future trajectories in this optimization, we must choose a single action to execute. One possible option is to add a constraint to ensure that the actions are the same for all particles, i.e. $u_i^{(j)} = u^{(j)}, \forall i = 1, \ldots, M, \ \ j = 0, \ldots, N$. We refer to this as *full consensus* over actions. Enforcing full consensus forces the optimization to find sequences of actions that will perform well, open-loop, across all sampled trajectories. However, in practice, control under uncertainty often involves re-optimization at every time step as in a model predictive control/receding horizon fashion. Furthermore, robotic systems typically employ filtering techniques to reduce uncertainty in state or dynamics online or otherwise have more information in the future not reflected in the open-loop control optimization problem. Full consensus fails to account for both this replanning and information gain, and thus may choose actions that are overly conservative. Another extreme is *one-step consensus*, wherein we only enforce the constraint that the action for the first action is shared. This ensures we still have a clear solution with regards to which action to select at the current time step, but allow subsequent actions to be different for each particle. Thus, this approach represents the least conservative approach, and is implicitly assuming that after one time step, we will have perfect state and dynamics knowledge. More generally, our framework allows choosing an arbitrary consensus horizon N_c, smoothly interpolating between the two extremes. A system designer can choose N_c in accordance with the replanning frequency or the expected rate of information gain for their particular system.

3.2. Combining Particle Representations with Sequential Convex Programming

The PMPC approach yields a large non-convex optimization problem. In order to approximately solve this problem efficiently, we employ sequential convex programming (SCP). SCP applies efficient, optimized solvers for convex optimization problems sequentially to locally optimize a non-linear optimization problem. At each step of SCP, we first form a convex approximation of the non-convex problem around a reference setting of the optimization variables, and then solve this convex problem to get a new reference point—while attempting to stay close to the previous reference point.

In the case of PMPC, the optimization variables are the state and action trajectories $\{\boldsymbol{x}_i, \boldsymbol{u}_i\}_{i=1}^M$ where $\boldsymbol{x}_i = (x_i^{(0)}, \ldots, x_i^{(N+1)})$ and $\boldsymbol{u}_i = (u_i^{(0)}, \ldots, u_i^{(N)})$. To convexify the problem at a particular setting of these optimization variables, we replace the dynamics with a linear approximation and the cost with a linear approximation of its non-convex part, yielding the convex optimization problem:

$$\begin{aligned}
\min_{\{\Delta \boldsymbol{x}_i, \Delta \boldsymbol{u}_i\}_{i=1}^M} \quad & \frac{1}{M} \sum_{i=1}^{M} \sum_{j=0}^{N} \left(\nabla_x c^{(j)} \Delta x_i^{(j)} + \nabla_u c^{(j)} \Delta u_i^{(j)} + \rho_x \|\Delta x_i^{(j)}\|_2^2 + \rho_u \|\Delta u_i^{(j)}\|_2^2 \right) \\
\text{s.t.} \quad & \Delta x_i^{(j+1)} = \nabla_x f_i \Delta x_i^{(j)} + \nabla_u f_i \Delta u_i^{(j)} \quad \forall j = 0, \ldots, N, \;\; i = 1, \ldots, M \\
& \bar{x}_i^{(j)} + \Delta x_i^{(j)} \in \mathbb{X}^{(j)} \quad \forall j = 0, \ldots, N, \;\; i = 1, \ldots, M \\
& \bar{u}_i^{(j)} + \Delta u_i^{(j)} \in \mathbb{U}^{(j)} \quad \forall j = 0, \ldots, N, \;\; i = 1, \ldots, M \\
& \Delta u_i^{(j)} = \Delta u_k^{(j)} \quad \forall j = 0, \ldots, N_c, \;\; i \neq k
\end{aligned} \tag{2}$$

where this optimization problem is written in terms of the deviations $\{\Delta \boldsymbol{x}_i, \Delta \boldsymbol{u}_i\}_{i=1}^M$ from the previous solution $\{\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i\}_{i=1}^M$, and the gradients are evaluated at $\{\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i\}_{i=1}^M$. Note that we add two terms to the cost to minimize deviation from the linearization point, as the approximation is only good for small deviations. While there are many strategies for limiting this deviation in each SCP iteration, we choose the quadratic penalties as they work well in practice and introduce only two hyperparameters ρ_x and ρ_u.

Algorithm 1 details the procedure for one planning iteration. Each iteration takes as input a set of particles, where each particle i has its own initial state and cost function. Note that in the case of a non-uniform belief over the particles, the relative weighting of the particles can be incorporated by scaling the particle's cost function appropriately. SCP alternates between convexification of the dynamics and cost functions, and subsequently solving the convex problem (2) by leveraging an off-the-shelf efficient convex solver. If the algorithm converges, we are guaranteed to have a locally near-optimal solution. We detect convergence by evaluating the norm of the change in the solution trajectory, as at a locally optimal solution, the call to `PMPC_convex` would not change the solution. When stopped prior to convergence, the solution is approximate up to the error in the linear approximation of the dynamics and the cost.

4. Discussion

4.1. Choice of Consensus Horizon

A key hyperparamter in PMPC is the consensus horizon N_c, and the range from *one-step* to *full* consensus has a large impact on the closed loop performance of the controller. Both extremes are common choices in the controls and decision making literature, with one-step consensus corresponding to the QMDP approximation of POMDPs (Littman et al., 1995) and full consensus corresponding to robust control. Yet for many systems, the optimal balance of practicality and conservatism may be achieved with a consensus horizon in between these extremes. One way to interpret the choice of

Algorithm 1 SCP PMPC

Function `PMPC_convex`$\big(\{x_i^{(0)},\ \bar{f}_i^{(j)},\ \nabla_x f_i,\ \nabla_u f_i,\ c_{i,cvx}^{(j)},\ \nabla_x c_{i,ncvx}^{(j)},\ \nabla_u c_{i,ncvx}^{(j)}\}_{i=1}^M,\ \rho_x,\ \rho_u,\ N_c\big)$

  $\{\Delta x_i^{(j)}, \Delta u_i^{(j)}\}_{i=1}^M \leftarrow$ solve the problem in (2)
  return $\{\Delta x_i^{(j)}, \Delta u_i^{(j)}\}_{i=1}^M$

end

Input: Initial states $\{x_i^{(0)}\}_{i=1}^M$, dynamics models $\{f_i\}_{i=1}^M$, solution guess $\{\boldsymbol{x}_i, \boldsymbol{u}_i\}_{i=1}^M$
Input: Hyperparameters $\rho_x, \rho_u, N_\text{c}$
Input: Solution tolerance ϵ

repeat

  $\{\bar{f}_i^{(j)}, \nabla_x f_i, \nabla_u f_i\}_{i=1}^M \leftarrow$ linearize the dynamics around the trajectory guess $\{\boldsymbol{x}_i, \boldsymbol{u}_i\}_{i=1}^M$
  $\{c_{i,\text{cvx}}^{(j)}, c_{i,\text{ncvx}}^{(j)}\}_{i=1}^M \leftarrow$ split the cost into the convex and non-convex parts
  $\{\nabla_x c_{i,\text{ncvx}}^{(j)},\ \nabla_u c_{i,\text{ncvx}}^{(j)}\}_{i=1}^M \leftarrow$ linearize the non-convex cost around $\{\boldsymbol{x}_i, \boldsymbol{u}_i\}_{i=1}^M$
  $\{\Delta x_i^{(j)}, \Delta u_i^{(j)}\}_{i=1}^M \leftarrow$ **call** `PMPC_convex` $\left(\{x_i^{(0)},\ \bar{f}_i^{(j)},\ \dots\}_{i=1}^M,\ \rho_x,\ \dots\right)$
  $\{\boldsymbol{x}_i, \boldsymbol{u}_i\}_{i=1}^M \leftarrow \{\boldsymbol{x}_i + \Delta\boldsymbol{x}_i, \boldsymbol{u}_i + \Delta\boldsymbol{u}_i\}_{i=1}^M$ update solution trajectory

until $\sum_{i=1}^M \sum_{j=0}^N \|\Delta x_i^{(j)}\| + \|\Delta u_i^{(j)}\| < \epsilon$
Output: solution trajectory $\{\boldsymbol{x}_i, \boldsymbol{u}_i\}_{i=1}^M$

consensus horizon is as an approximation of the information gain dynamics: uncertainty is constant for N_c steps, after which it drops immediately to zero when full information is revealed. By enforcing control consensus for N_c steps, PMPC is assuming that the uncertainty will remain constant for this time period, and so much choose actions that work for all possibilities. After the consensus horizons, controls can be tailored to each particle, which is only possible when the true system is revealed. In robotics systems with online filtering and belief updating, it is often the case that full consensus will yield plans that are over-conservative. However, the other extreme of one-step consensus can yield plans that are under-conservative. In fact, as we show in the following theorem, for any consensus horizon H, we can design a system for which $N_\text{c} = K$ is optimal but any shorter consensus horizon is arbitrarily suboptimal.

Theorem 1 *For every consensus horizon less than the planning horizon, $1 < N_c < N$, there exists a dynamical system with uncertainty for which the plan with $N_c^- = N_c - 1$ is arbitrarily suboptimal.*

The proof is available in the appendix, available at (Dyro et al., 2020).

4.2. Particle Model Predictive Control vs Certainty Equivalent Model Predictive Control

A baseline for control under uncertainty is to assume *certainty equivalence* (CE), i.e. choose a set of actions assuming a nominal model is exactly correct. With an exception of system with linear dynamics and Gaussian state uncertainty (Bertsekas, 2012), this technique is sub-optimal, but nevertheless is easy to implement and commonly used in robotics. CE control exists as a special case of PMPC where we use only a single particle corresponding to the nominal setting of all uncertain quantities. While CE control may seem appealing as a simpler, easier to implement solution, it is significantly complicated when dealing with multimodal uncertainties for which the mean may not be an accurate representation. For example, with multimodal sensing uncertainty, planning assuming the robot is at the mean state may lead to actions that are unsuitable for all possible scenarios.

PMPC allows directly translating arbitrary uncertainty representations on state, dynamics, and cost function to the controller, sidestepping the often complex decision of reducing an uncertain system to a certain one for CE control. In our experiments, we demonstrate that PMPC outperforms CE control on common sources of uncertainty, both uni- and multi-modal.

5. Related Work

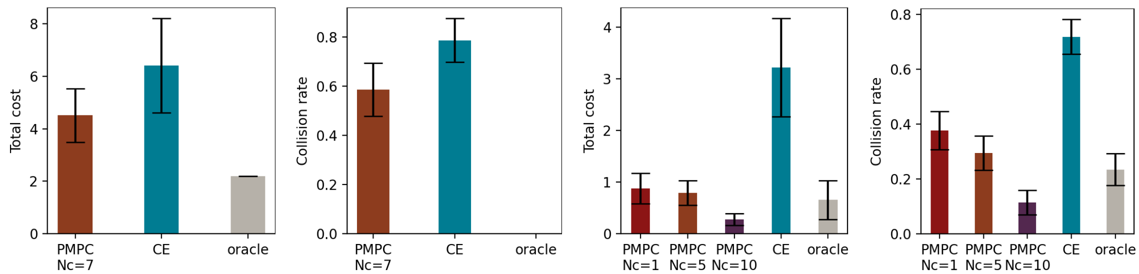
This work addresses the problem of optimal control with uncertainty and so most broadly belongs to the fields of stochastic and robust optimal control, each of which have long histories. However, in the context of arbitrary uncertainty considered in this work, the most common approaches model the impact of uncertainty via assuming worst-case disturbances (as in, for example, (Kothare et al., 1996; Khargonekar et al., 1988). These approaches are typically overly conservative or too computationally expensive.

These robust methods, in which probabilistic uncertainty are mapped into set-theoretic ones, may overestimate the impact of improbable disturbances. Thus, work over the last several decades has focused on distributional approximations to uncertainties, typically approximating the disturbance distribution as a Gaussian. These approaches, such as those of Cinquemani et al. (2011); Deisenroth and Rasmussen (2011); Deisenroth (2012), yield tractable solutions, but are often poor approximations for the arbitrary uncertainty that arises in robotic systems, such as binary actuator failure. Moreover, works which model the uncertainty as additive iid noise, like Kantas et al. (2009); Deisenroth and Rasmussen (2011), do not take into account the temporal correlation of the model uncertainty. For example, if a non-varying dynamics parameter is unknown, the impact of this parameter will be different (and typically result in a wider distribution of outcomes) than if the same parameter is re-sampled at every time step. We refer the reader to McHutchon (2015) for a more detailed description of this.

The approach considered in this work, finite sampling approximation to the arbitrary uncertainty in the state and model dynamics have been suggested in multiple contexts De Villiers et al. (2011) Sehr and Bitmead (2017), combined with partial state observability by coupling the planning with a particle filter Stahl and Hauth (2011) and applied to dynamical systems Shimada and Nishida (2014). However, despite the application of a particle approximation, the resulting control optimization problem remains nonlinear in the presence of nonlinear dynamics. Where an optimization technique is suggested, it typically involves random search, which scales poorly as the number of control dimensions increases and provides no mechanism of estimating the quality of the solution.

Iterative importance sampling control schemes such as that of Williams et al. (2017) have seen application in setting with model uncertainty (Chua et al., 2018; Abraham et al., 2020; Arruda et al., 2017). These methods do not rely on the gradient of the cost function (or dynamics) for optimization, and are strictly sampling-based. These methods have seen widespread application in learning-based control due to their good performance relative to their simplicity. Moreover, part of their popularity is attributable to few good alternatives capable of handling arbitrary uncertainty, which we hope to address with this work. Fundamentally, better performance is achievable with comparable computational complexity by considering available gradients, as we show in our experiments.

Many works, like us, more explicitly consider the optimization technique in the design of particle-based methods. Blackmore et al. (2010) assumes linear dynamics and provides a Mixed Integer Linear Program (MILP) formulation to incorporate chance constraints and hybrid dynamics. Likewise, Calafiore and Fagiano (2012) consider linear dynamics and focus on worst case particle optimization to provide a tight optimality bound for the particle solution under arbitrary uncertainty as the number of particles increases. While these methods allow better handling of chance constraints than our approach, they do not address non-linear dynamics beyond hybrid linear systems and are thus quite limited. Most closely related to this work is (Wytock et al., 2017) where



(a) Free-flyer (b) Free-flyer (c) Planar quadrotor (d) Planar quadrotor

Figure 1: Total cost and collision rate for the free-flyer and the planar quadrotor under **model uncertainty**.

the weighted per particle cost is optimized in the context of a dynamical system using convex optimization with an explicit consensus horizon on control. However, the work again assumes linear dynamics and does not fully discuss the consensus horizon choice, suggesting it is set to 1.

6. Experiments

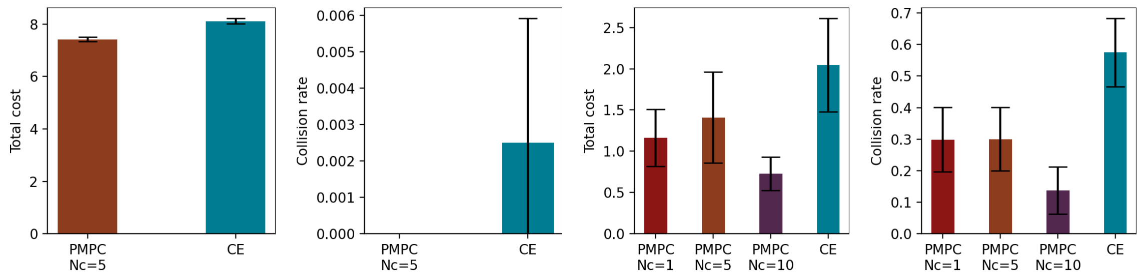
We demonstrate the utility of PMPC by evaluating its performance on two nonlinear systems in three settings each, chosen to highlight distinct sources of uncertainty arising in robotics problems. Specifically, we use (1) a 6-D planar quadrotor (2-D action space), a common system for benchmarking highly dynamic control and reinforcement learning algorithms (Ivanovic et al., 2019; Gillula et al., 2010; Singh et al., 2017) and (2) a 6-D planar free-flyer, a free floating spacecraft constrained to a plane with actuation through gas thrusters and a reaction wheel (9-D action space) (Lew et al., 2020).

For each of these systems, we consider three sources of uncertainty that are prevalent throughout robot autonomy: (i) changing system dynamics within an otherwise known dynamics model (e.g. actuator failures or external effects), (ii) state uncertainty, and (iii) epistemic uncertainty within black-box learning-based control (model-based reinforcement learning). In each of these settings, we execute a task corresponding to locomotion to a desired position while avoiding obstacles. We encode this task using a cost function which generally takes the form of a quadratic cost on position to encourage movement towards the goal, with additive penalty for collision with obstacles which is linear in the amount violation. Note that the nonconvexity of the free space makes this cost function nonconvex. We also include experiments investigating the computational complexity of the method for varying numbers of particles, showing the approach is feasible in real-time.

Because of the wide variety of test environments, we exclude details of each environment from the body of the paper. The details for all environments, including details of the design of the particle dynamics in the control scheme, are available in the appendix (Dyro et al., 2020). This appendix also includes additional experimental results. Throughout all figures in this section, confidence intervals are 95%.

6.1. Dynamics Uncertainty

Dynamics uncertainty is commonplace in robotics. Robotics systems may evolve over time stochastically, e.g. with nonzero chances of actuators failing, or temporally evolving external factors like wind disturbances. We use the two systems to investigate the performance of PMPC in these characteristic settings of dynamics uncertainty.



(*a*) Free-flyer (*b*) Free-flyer (*c*) Planar quadrotor (*d*) Planar quadrotor

Figure 2: Total cost and collision rate for the free-flyer and the planar quadrotor under **state uncertainty**.

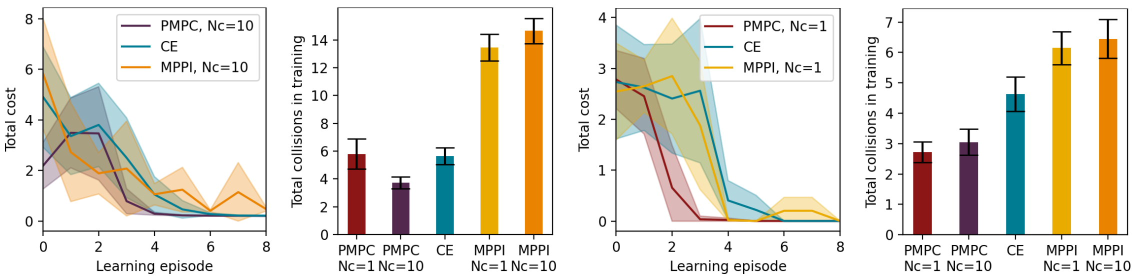
For the free-flyer problem, we consider a scenario in which the robot aims to regulate to a position near a wall, mimicking a docking scenario. For this system, each thruster has a random failure probability at each time step. The current failure status (a discrete random variable) is inferred online by a recursive Bayes filter. We compare to a certainty equivalent (CE) formulation that acts with respect to the MAP failure status estimate.

For the planar quadrotor, we assume a wind disturbance modeled by an Ornstein-Uhlenbeck process in both spatial dimensions. This process is, roughly, a Brownian motion process with a term that pulls the process toward 0 and thus captures the stochastic but temporally correlated nature of wind gusts. In this problem, the quadrotor aims to navigate a narrow passage while avoiding collision. The CE baseline assumes the current wind remains constant.

Results for both systems can be seen in Figure 1. In both experiments, we also compare to an oracle baseline which has perfect knowledge of future dynamics changes. For the free-flyer system, we plot only $N_c = 7$ as results were consistent across varying consensus horizons. For the planar quadrotor, we visualize performance for $N_c = 1, 5, 10$. In our experiments, all PMPC approach outperformed the CE approach, and avoided collisions substantially more frequently. Interestingly, PMPC with $N_c = 10$ actually outperformed the oracle model. While the oracle model has perfect knowledge of the time evolution of parameters governing stochastic disturbances *within the finite planning horizon*, it fails to account for sample values beyond it. Thus, the conservatism added by the consensus horizon improves performance due to additional robustness to these stochastic disturbances.

6.2. State Uncertainty

A common source of uncertainty in robotics comes from partial observability of the system's state. Typically, robotic systems employ online filtering to estimate their current state, for example, by using a particle filter (Thrun et al., 2005). We test how well SCP PMPC can address this form of uncertainty on both systems. In both cases, we assume we only have uncertainty on position, initialized as a unimodal distribution of particles. This distribution is updated online using a particle filter given noisy observations of distance from four laser range finder sensors aligned with cardinal directions in the vehicle frame. As the true observation noise is often not known exactly, we choose a different noise covariance for the particle filter based on performance of the downstream controller. These particles are directly fed in to the SCP PMPC controller. For the CE controller, we plan assuming the state is the expected position of the particles. In both settings, we evaluate performance over scenarios where the true position is sampled from initial belief over position.



(a) Free-flyer (b) Free-flyer (c) Planar quadrotor (d) Planar quadrotor

Figure 3: Learning curves (cost per iteration) and total number of collisions during training for the free-flyer and the planar quadrotor in a model-based reinforcement learning setting.

Results for this setting are plotted in Figure 2. Again, we see uniformly better performance for the PMPC controller compared to the CE controller. We note that while the cost difference between the PMPC and CE controller for the free-flyer appear close,the difference in cost is highly significant, and the majority of the cost is can not be reduced even with perfect information (due in part to the rarity of collisions).

6.3. Learning for Control

A third, and increasingly common source of uncertainty is robotics arises from learned components. Any learned component has *epistemic uncertainty*, reflecting uncertainty in the underlying function as opposed to *aleatoric*, or irreducible, uncertainty which is a feature of the environment. Characterization of this epistemic uncertainty has led to substantial improvements in learning-based control and reinforcement learning due to better robustness with respect to unknown dynamics and better exploration (Chua et al., 2018). A common tool to quantify this uncertainty is using deep ensembles (Lakshminarayanan et al., 2017), wherein several neural networks are trained on the same data but starting from different initializations and regularized to different points. Following the problem setting of (Chua et al., 2018), we consider the problem of control with a learned dynamics model. Specifically, we consider a model-based RL framework, as the initial low-data regime highlights the need to factor in epistemic uncertainty.

We train D neural network dynamics models between every episode on the state, action, next-state tuples collected so far. We minimize a squared ℓ_2 loss, with each network regularized to it's random initialization, following methods proposed by (Osband et al., 2018; Pearce et al., 2020). We use a fixed learning rate of 10^{-3} and train until convergence. During the episode, we use each network as the (stationary) dynamics for a single particle in the PMPC controller. We compare against a CE controller that uses only the mean prediction of the ensembles (ignoring epistemic uncertainty) as well as an MPPI controller (Williams et al., 2017) which factors in this uncertainty but is a sampling-based, gradient free method (as used in Chua et al. (2018)).

Our results are visualized in Figure 3. Figures (a) and (c) show learning curves for the reinforcement learning process: they show the cost per episode during learning. There are several things to note in these figures. First, both the CE and PMPC controller outperform the MPPI controller, showing improved performance as a result of gradient-based controllers as opposed to stochastic, sampling-based schemes. Second, the performance improvement of PMPC versus CE methods. Although the collisions in training are relatively close, the training curves for both systems show convergence 2-4 episodes before the CE approach. Additionally, the MPPI controller shows sub-optimality in later episodes whereas the both the CE and PMPC controllers achieve consistent performance. This performance variation is likely due to the stochastic sampling-based controller.

6.4. Computational Complexity

Figure 4 shows time per iteration for a varying number of particles, for $N_c = 1, 5, 10$. The plots were generated on a Intel(R) Core(TM) i7-8559U CPU @ 2.70GHz. The slope of this log-log plot is approximately 1, showing approximately linear computational complexity in the number of particles. This is due to the utilization of sparse quadratic programming solvers. Moreover, interestingly, the larger consensus period does not have a substantial impact on computational complexity until very large numbers of particles are used ($\approx$ 1000), where there is an approximately 1.5$\times$ performance difference for $N_c = 5, 10$ versus $N_c = 1$. The results show for an intermediate number of particles (e.g. approximately 20), operational frequencies of around 50 are possible even with off-the-shelf solvers and standard consumer CPUs. With GPU acceleration, we anticipate operational frequencies of 100Hz with up to 50 particles is achievable in the near term.

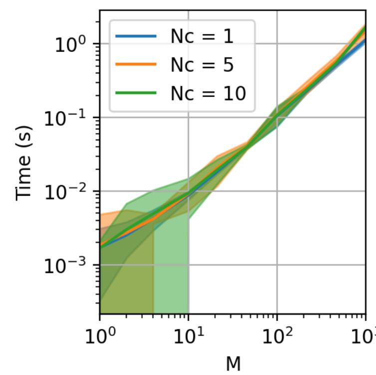


Figure 4: Time per iteration of the nonlinear particle MPC scheme for varying number of particles, M.

We highlight that the Monte Carlo methods we develop in this paper are currently becoming computationally feasible in real time due to advances in computational hardware for parallel processing and sparse solvers. Thus, we believe that effective design of Monte Carlo-based methods as opposed to imprecise approximations discussed in the introduction will be a fruitful avenue of research for uncertainty characterization in control in coming years.

7. Conclusions

In this work we have presented a framework for particle-based nonlinear model predictive control enabling effective control under uncertainty. We have shown the performance of this method for two different robot systems, in three different settings. Across all settings, the performance of the PMPC framework outperformed the certainty equivalent controller baseline. In the model-based RL (MBRL) problem setting, PMPC substantially outperforms uncertainty-aware MPPI (Williams et al., 2017), which has been shown to substantially improve MBRL relative to CE methods (Chua et al., 2018). Finally, we have investigated the computational complexity of the proposed approach and found that the method is capable of real time operation, and will become increasingly capable for dynamic real time applications as computational hardware and sparse solvers improve. Thus, further algorithmic improvements in particle-based control methods represent a promising direction of future research for control under uncertainty.

Acknowledgments

Some of the computing for this project was performed on the Sherlock cluster. We would like to thank Stanford University and the Stanford Research Computing Center for providing computational resources and support that contributed to these research results. James Harrison was supported in part by the Stanford Graduate Fellowship and the National Sciences and Engineering Research Council of Canada (NSERC).

References

Ian Abraham, Ankur Handa, Nathan Ratliff, Kendall Lowrey, Todd D Murphey, and Dieter Fox. Model-based generalization under parameter uncertainty using path integral control. *IEEE Robotics and Automation Letters (R-AL)*, 2020.

Ermano Arruda, Michael J Mathew, Marek Kopicki, Michael Mistry, Morteza Azad, and Jeremy L Wyatt. Uncertainty averse pushing with model predictive path integral control. In *IEEE-RAS International Conference on Humanoid Robots*, 2017.

Dimitri P Bertsekas. *Dynamic programming and optimal control*. Number 1. 4 edition, 2012.

Lars Blackmore, Masahiro Ono, Askar Bektassov, and Brian C. Williams. A probabilistic particle-control approximation of chance-constrained stochastic predictive control. *IEEE transactions on Robotics*, 26(3):502–517, 2010. Publisher: IEEE.

Giuseppe C. Calafiore and Lorenzo Fagiano. Robust model predictive control via scenario optimization. *IEEE Transactions on Automatic Control*, 58(1):219–224, 2012. Publisher: IEEE.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, pages 4754–4765, 2018.

Eugenio Cinquemani, Mayank Agarwal, Debasish Chatterjee, and John Lygeros. Convexity and convex approximations of discrete-time stochastic control problems with constraints. *Automatica*, 47(9):2082–2087, 2011. Publisher: Elsevier.

Johan Pieter De Villiers, S. J. Godsill, and S. S. Singh. Particle predictive control. *Journal of statistical planning and inference*, 141(5):1753–1763, 2011. Publisher: Elsevier.

Marc Deisenroth and Carl E Rasmussen. PILCO: A model-based and data-efficient approach to policy search. *International Conference on Machine Learning (ICML)*, 2011.

Marc Peter Deisenroth. Learning to control a low-cost manipulator using data-efficient reinforcement learning. *Robotics: Science and Systems (RSS)*, 2012.

Robert Dyro, James Harrison, Apoorva Sharma, and Marco Pavone. Particle MPC for uncertain and learning-based control (extended version). *TODO*, 2020.

Jeremy H Gillula, Haomiao Huang, Michael P Vitus, and Claire J Tomlin. Design of guaranteed safe maneuvers using reachable sets: Autonomous quadrotor aerobatics in theory and practice. *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

Boris Ivanovic, James Harrison, Apoorva Sharma, Mo Chen, and Marco Pavone. BARC: Backward reachability curriculum for robotic reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

Nikolas Kantas, J. M. Maciejowski, and A. Lecchini-Visintini. Sequential Monte Carlo for model predictive control. In *Nonlinear model predictive control*, pages 263–273. Springer, 2009.

P.P. Khargonekar, I.R. Petersen, and M.A. Rotea. H/sub infinity /-optimal control with state-feedback. *IEEE Transactions on Automatic Control*, 33(8):786–788, August 1988. ISSN 00189286. doi: 10.1109/9.1301. URL http://ieeexplore.ieee.org/document/1301/.

Mayuresh V. Kothare, Venkataramanan Balakrishnan, and Manfred Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, 1996. Publisher: Elsevier.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Neural Information Processing Systems (NeurIPS)*, 2017.

Thomas Lew, Apoorva Sharma, James Harrison, and Marco Pavone. Safe model-based meta-reinforcement learning: A sequential exploration-exploitation framework. *arXiv:2008.11700*, 2020.

Michael L Littman, Anthony R Cassandra, and Leslie Pack Kaelbling. Learning policies for partially observable environments: Scaling up. In *International Conference on Machine Learning (ICML)*. 1995.

Andrew James McHutchon. *Nonlinear modelling and control using Gaussian processes*. PhD thesis, Cambridge University, 2015.

Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, pages 8617–8629, 2018.

Tim Pearce, Felix Leibfried, and Alexandra Brintrup. Uncertainty in neural networks: Approximately bayesian ensembling. In *Artificial Intelligence and Statistics (AISTATS)*, 2020.

Martin A. Sehr and Robert R. Bitmead. Particle model predictive control: Tractable stochastic nonlinear output-feedback MPC. *IFAC-PapersOnLine*, 50(1):15361–15366, 2017. Publisher: Elsevier.

Kento Shimada and Takeshi Nishida. Particle filter-model predictive control of quadcopters. In *Proceedings of the 2014 International Conference on Advanced Mechatronic Systems*, pages 421–424. IEEE, 2014.

Sumeet Singh, Anirudha Majumdar, Jean-Jacques Slotine, and Marco Pavone. Robust online motion planning via contraction theory and convex optimization. *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

Dominik Stahl and Jan Hauth. PF-MPC: Particle filter-model predictive control. *Systems & Control Letters*, 60(8):632–643, 2011. Publisher: Elsevier.

Bartolomeo Stellato, Goran Banjac, Paul Goulart, Alberto Bemporad, and Stephen Boyd. OSQP: An operator splitting solver for quadratic programs. *Mathematical Programming Computation*, pages 1–36, 2020. Publisher: Springer.

Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT Press, 2005.

Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A Theodorou. Information theoretic mpc for model-based reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

Matt Wytock, Nicholas Moehle, and Stephen Boyd. Dynamic energy management with scenario-based robust MPC. In *2017 American Control Conference (ACC)*, pages 2042–2047. IEEE, 2017.