Robot Manipulation Under Uncertainty

Lara Brudermüller

Exeter College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

October 2025

Acknowledgements

I am forever grateful to my advisor, Nick Hawes, for his continuous support, guidance, and patience throughout my PhD, and for giving me endless freedom, not only in my research but also in shaping my own path throughout this PhD. Thank you for always believing in me, even when I doubted myself, and for helping me grow into the researcher I am today. I would also like to thank Marc Toussaint, who over the years became a second advisor to me. Thank you for welcoming me into your group at TU Berlin, even beyond the time of my research stay, for your constant openness to discuss ideas and provide thoughtful feedback, and for your contagious enthusiasm for robotics, which has deeply inspired me.

My heartfelt thanks go to my wonderful colleagues in Oxford, who have supported me in countless ways throughout my PhD. The "GOALS-gang" has been an incredible group of people to work and laugh with, and I am grateful for the inspiring discussions, collaborations, and fun times we shared. A special thanks to Bruno, Clarissa, Anna, Charlie, Marc, Matt, Branton, Alex R., Alex St., Michal, and Roland, who have been my colleagues for most of my PhD and who made this journey so much more enjoyable. Clarissa and Anna – thank you for being my "GOALS-girls" and so much more than colleagues. I am also beyond lucky that Efimia joined the Oxford Robotics Institute around the same time as I did and has since become a dear friend – I admire all three of you for being such strong female role models in robotics and AI. I am also thankful for my other (former) colleagues at the ORI, especially Jack Collins and Jun Yamada, for many inspiring discussions, and Ingmar Posner and Ioannis Havoutis, who assessed my academic progress at various stages and offered valuable feedback.

Outside Oxford, I would like to thank my colleagues in Berlin – Cornelius, Valentin, Sayantan, Shiping, Hongyou, Svetlana, Ilaria, Pia, Khaled, Akmaral, Christoph, Omar, Denis, Eckart and Wolfgang for being truly welcoming and supportive throughout my time in Berlin. A special thanks goes to Cornelius, with whom I shared an office and who has become a good friend, for the great scientific discussions and equally great non-scientific gossip; to Pia, for always believing in me, supporting me, and being my sports buddy; and to Ilaria, for making me feel welcome from day one.

I am also deeply grateful to my mentors from my internship at the Robotics and AI Institute in Boston. Thank you to Simon and Preston for being incredible

mentors during and beyond my internship, to Annika for your unwavering support and mentorship to this day, and to Lesley, Tarik and Ali for making both my internship and my time in Boston so much fun. Finally, thank you to Guillaume for collaborating with me on our journal paper, and for sharing the joys and frustrations of a long publication process.

Beyond academia, I owe immense gratitude to my friends outside work, who have been incredibly supportive throughout my PhD. To my friends in Oxford – Peter, Adam, Rory, Tristram, Sasha, Annie, Hannah, and many more – thank you for all the fun times we shared and for always being there for me. I will fondly remember our climbing and sailing trips, tennis matches, and the many dinners, constantly being spoiled with good food. And finally, to my closest friends in Germany – Marie-Sophie, Sarah, Florian, Jan, Tom, Moritz, Alex, Anne, Louissa, Lucas, Nikolai, Jeany and Luca – thank you for your endless encouragement, for understanding the sacrifices that I have made for this PhD, and for constantly reminding me that I am the luckiest person to have friends like you in my life.

I would not be where I am today without the unwavering support of my family. First and foremost, thank you to my parents for your endless love and encouragement, and for always being there when I needed you most. You've always supported me in pursuing my dreams, and I'm endlessly grateful for everything you've done to help me get here. To my twin brother, Tobias – thank you for being the strong pillar of support that you are in my life and for sharing this PhD journey with me. And to my grandparents, for always making me feel loved and cherished.

However, most importantly, my forever deepest gratitude goes to my partner, Julius, who has not only been my rock throughout this PhD but also my greatest source of inspiration and happiness in life. Sharing this passion for robotics with you by writing papers[1] together, has made this journey even more special. I cannot express how thankful I am for your endless support, patience, and understanding, for always believing in me, and for giving me the energy to always keep going. You have made this entire experience so much more meaningful and enjoyable, even across distance, and I feel incredibly lucky to have you by my side. I cannot wait to see what the future holds for us.

[1]Some call them "love papers".

Abstract

Robots in unstructured environments must perceive, plan, and act under significant uncertainty. Unstructured environments are characterised by cluttered, dynamic, and partially observable settings, as found in homes, offices, and public spaces. To operate effectively in such environments, robots must frequently interact with unknown objects; making and breaking contact as they push, deform, or reorient them during manipulation. Anticipating the outcome of these contact-rich interactions is critical for robot autonomy but remains challenging due to the discontinuous nature of contact dynamics and the uncertainty arising from unmodelled physical properties of the environment. This thesis addresses the problem of controlling robots to exploit contact as a purposeful means of manipulation. In the first part, we formulate the problem of contact-rich manipulation as a model predictive control problem and show that stochastic optimisation in combination with informative priors and learned proposal distributions enables efficient exploration of the contact space in real-time without relying on local gradients or discretisation of the contact space. The ability to replan at high frequency allows the robot to adapt online to changing dynamics, making it *implicitly* robust to uncertainty. In the second part of this thesis, we extend this framework to *explicitly* account for uncertainty in the robot's state and environment. We introduce chance constraints to ensure probabilistic constraint satisfaction and develop a sampling-based approximation that enables efficient evaluation of these constraints within the model predictive control loop. Finally, we explore belief space control through contact, enabling the robot to actively reduce uncertainty in its environment by seeking out informative contact interactions. This allows the robot to not only react to uncertainty but also to explicitly reason about it and take actions to reduce it. Together, these approaches provide a coherent framework that combines reactivity with principled decision-making under uncertainty, enabling efficient and robust robot control strategies that can actively manage and mitigate uncertainty in real-time. We demonstrate the effectiveness of these methods in dynamic robot handover tasks and contact-rich manipulation with a robot manipulator, as well as a quadruped robot with an arm.

Keywords: Trajectory Optimisation, Stochastic Optimisation, Model Predictive Control, Stochastic Dynamics, Robust Control, Contact-Rich Manipulation

Contents

1 Introduction

Contents

Robots are increasingly deployed in unstructured, dynamically changing environments such as homes, warehouses, and care facilities, where they must interact safely and intelligently with both humans and objects. Unlike in controlled industrial settings, these environments present variability, ambiguity, and incomplete information. Operating effectively in these settings requires the ability to *reason about uncertainty*. Consider a robot tasked with unloading a dishwasher in a home environment. Almost certainly, it will encounter a variety of novel objects whose physical properties (e.g., mass, friction) cannot be fully known without interaction. However, any such interaction, whether visual or physical, yields only noisy and incomplete sensory information. Visual observations may be compromised by occlusions in a fully packed dishwasher or by challenging lighting conditions, while force or tactile feedback can be confounded by the robot's own dynamics and the compliance of the manipulated objects. As a result, these partial and

uncertain observations rarely provide a complete or unambiguous representation of the world. Besides, the robot's own actions can also increase the uncertainty about the environment (e.g., by moving objects or causing them to fall). Moreover, domestic environments are inherently dynamic and subject to change, as humans, and potentially other robots, simultaneously interact with the space in unpredictable ways. These interactions introduce further uncertainty in the environment's state and evolution over time, essentially turning them into stochastic systems. Developing methods that can handle these forms of uncertainty is essential for enabling robust, adaptive, and safe robot behaviour in open-world scenarios such as the one described above. This thesis explores different approaches to handling these forms of uncertainty in order to achieve reliable and efficient robot behaviour in such stochastic environments. Together, these approaches can be integrated into a unified framework for *robot motion planning and control under uncertainty*, enabling more robust and adaptable behaviour.

Sources of Uncertainty With the goal of contributing towards enabling robots to operate reliably and efficiently in these stochastic environments, the first step is to characterise the sources of uncertainty that this thesis aims to address. The sources of uncertainty can be broadly categorised into two types: *aleatoric* and *epistemic* uncertainty (Sullivan, 2015). *Aleatoric* uncertainty, also known as statistical uncertainty, arises from inherent randomness or variability in the environment or system. This type of uncertainty is irreducible, meaning that no amount of additional information can eliminate it. Examples of aleatoric uncertainty in robotics include sensor noise, actuator noise, and environmental variability. *Epistemic* uncertainty, on the other hand, stems from a lack of knowledge or information about the environment or system. This type of uncertainty is reducible, meaning that it can be mitigated through additional data or learning. Examples of epistemic uncertainty in robotics include incomplete models of the environment, unknown object properties, and unmodelled dynamics. In the context of robot manipulation, both types of uncertainty are prevalent and should be addressed to achieve robust

and reliable performance. For instance, when a robot attempts to grasp an object, it may encounter aleatoric uncertainty due to sensor noise affecting its perception of the object's position and orientation. As a result, the robot might slightly misalign its gripper, causing unstable or failed grasps even if the object's properties are perfectly known. Simultaneously, it may face epistemic uncertainty if it lacks knowledge about the object's mass distribution or frictional properties, which can significantly impact the success of the grasp. In this case, even if the gripper is placed accurately, the robot may not apply sufficient force to prevent slippage, or it may use excessive force that risks damaging the object. These two types of uncertainty therefore affect the robot's behaviour in different ways: aleatoric uncertainty primarily leads to variability and unreliability in execution, while epistemic uncertainty leads to poor decision-making due to incomplete or incorrect models. To operate effectively, a robot must handle both. In this thesis, we address aleatoric uncertainty *implicitly* through high-frequency replanning, and epistemic uncertainty *explicitly* by modelling and reasoning about unknowns to ensure probabilistic constraint satisfaction and informed decision-making in stochastic environments.

Model-based vs. model-free The methods discussed in this thesis are predominantly *model-based*, meaning that they assume access to a model of the environment and robot dynamics. In the cases of implicit uncertainty handling, this model is assumed to be deterministic, whereas in the cases of explicit uncertainty handling, the model is assumed to be probabilistic, i.e., sufficiently capturing the stochasticity in the environment. More specifically, in the case of a probabilistic model, we only assume *sampling-based* access to the model, meaning that we can sample from the model to generate possible instantiations of the uncertainty, but we do not assume access to a closed-form representation of the uncertainty, such as a Gaussian distribution. This is a realistic assumption in many robotics applications, where the true distribution of the uncertainty is often unknown or difficult to model accurately. We also note that this also allows for data-driven models, such as learned dynamics models, to be used within the developed methods, as long as they

can provide sampling-based access to the uncertainty. In contrast to model-free approaches, which learn a policy directly from data without an explicit model of the environment, model-based approaches can leverage prior knowledge about the environment and robot dynamics to inform decision-making.

Focus on Contact-Rich Manipulation While the approaches presented in this thesis are applicable to a wide range of robotic tasks and robot embodiments, the main focus of this thesis is on robot manipulation tasks that involve physical interaction with objects and the environment. Such contact-rich tasks are particularly challenging due to the complex and discontinuous dynamics that arise from *making and breaking contact*. While some methods address this challenge by smoothing the contact dynamics, thereby enabling the use of gradient information, these approximations may limit accuracy in highly discontinuous settings. In contrast, the methods developed in this thesis avoid such approximations by relying on sampling-based techniques that explore the contact space directly without gradient information. This is particularly important when reasoning not only about individual contact modes, i.e., points in time where the robot is in contact with an object, but also about transitions between different contact modes over longer, potentially variable, time horizons. Moreover, contact-rich manipulation is inherently uncertain (Yu et al., 2016), both because contact dynamics are difficult to model accurately and because object properties such as shape, mass distribution, friction, and compliance are often unknown. These challenges are central in real-world applications like household chores, assembly, and service robotics, where robots must handle diverse objects in unstructured settings.

1.1 Contributions

This thesis contributes methods that enable robots to operate reliably and efficiently in stochastic environments, in particular in the context of contact-rich manipulation. The contributions span two complementary perspectives on uncertainty handling:

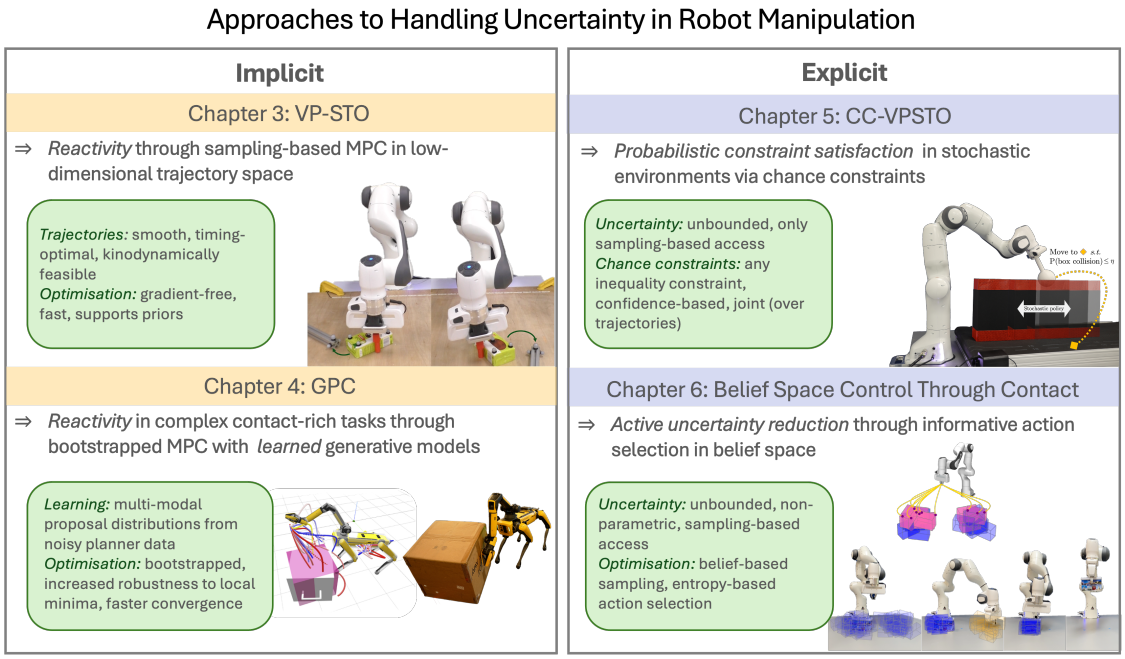


Figure 1.1: Overview of the contributions of this thesis for handling uncertainty in robot manipulation. The developed methods can handle uncertainty in robot motion planning and control, both *implicitly* through high-frequency replanning, as well as *explicitly* through modelling and reasoning about uncertainty. The methods are evaluated in the context of robot manipulation tasks.

1. *Implicit approaches*, where high-frequency replanning and fast feedback loops provide robustness by adapting online to uncertain and changing dynamics.

2. *Explicit approaches*, that model and reason about uncertainty in order to provide additional robustness and performance benefits, through improved decision-making, and active information gathering.

Together, these methods enable robots to act reliably and efficiently in stochastic environments. To address the discontinuities that arise from making and breaking contact, all methods developed in this thesis are *sampling-based*, meaning that they do not rely on gradient information and can therefore explore the contact space without additional approximations and smoothing operations.

We provide a high-level overview of the contributions made in each chapter of this thesis in Figure 1.1. Since this is an integrated thesis, each of these chapters corresponds to a publication or a submitted manuscript, as summarised in Table 1.1. To improve coherence of presentation, the chapters are ordered thematically rather

Chapter	Paper reference
Chapter 3	Jankowski*, J., Brudermüller*, L., Hawes, N., and Calinon, S. (2023). VP-STO: Via-point-based stochastic trajectory optimization for reactive robot behavior. *IEEE International Conference on Robotics and Automation (ICRA)*.
Chapter 4	Brudermüller, L., Hung, B., Zhu, X., Wang, J., Hawes, N., Culbertson, P., and Le Cleac'h, S. (2025). Generative models from and for sampling-based MPC: A bootstrapped approach for adaptive contact-rich manipulation. Manuscript under review at *IEEE Robotics and Automation Letters*.
Chapter 5	Brudermüller, L., Berger, G. O., Jankowski, J., Bhattacharyya, R., Calinon, S., Jungers, R. M., and Hawes, N. (2025). CC-VPSTO: Chance-Constrained Via-Point-Based Stochastic Trajectory Optimisation for Online Robot Motion Planning under Uncertainty. Manuscript under review at *The International Journal of Robotics Research*.
Chapter 6	Brudermüller, L., Jankowski, J., Toussaint, M., and Hawes, N. (2025). Touch-Based Object Localisation with Spatially-Aware Belief Entropy Estimation. Manuscript under review at *IEEE International Conference on Robotics and Automation (ICRA)*.

Table 1.1: Publications that form the basis of this thesis. Authors marked with an asterisk contributed equally.

than strictly chronologically. In each of these *paper chapters*, we provide a brief summary of the contributions and repeat Table 1.2 to situate the work within the broader context of robot motion planning and control under uncertainty. If not noted otherwise, we include supplementary material, as well as an additional discussion of limitations and future work, at the end of each chapter. Citations in the main body of this thesis refer to the bibliography at the end of this document, whereas citations in the individual chapters refer to the bibliographies included in the respective manuscripts.

While Figure 1.1 highlights the overarching structure and main ideas, it is complemented by Table 1.2, which compares the chapters and the respective contributions made in each chapter more systematically along several key dimensions:

- **Theme:** the main focus of the chapter, summarising the approach's primary strategy for handling uncertainty. This can be either *Reactivity*, where the approach relies on frequent replanning to adapt to changes; *Learning for Reactivity*, where the approach incorporates learning mechanisms to enhance its reactive capabilities; *Robustness*, where the approach explicitly models

uncertainty to ensure reliable performance; or *Exploration*, where the approach actively seeks out information to reduce uncertainty.

- **Uncertainty Handling:** whether the respective approach handles uncertainty *implicitly* or *explicitly*. Implicit handling of uncertainty typically relies on reactivity and frequent replanning, whereas explicit handling of uncertainty involves modelling and reasoning about uncertainty within the planning or control process.
- **Prior Knowledge of Uncertainty:** whether the approach assumes prior knowledge of the uncertainty, or whether it can operate without any prior knowledge. This dimension is tightly coupled with the previous one, as implicit handling of uncertainty typically does not require prior knowledge, whereas explicit handling does.
- **Environment:** whether the approach assumes deterministic or stochastic environments. Note that, even when assuming a deterministic environment, the method may still be able to handle uncertainty implicitly through reactivity.
- **Method:** the underlying method used for planning or control. While all approaches are sampling-based, they differ in the specific method used, such as Model Predictive Control (MPC) or belief-space control.
- **Control Problem:** the specific optimal control problem addressed by the approach, such as cost minimisation, chance-constrained cost minimization, or information gain maximisation.

Beyond the developed methods and experimental validations, the theoretical insights presented in this thesis should serve as a foundation for future research on robust and adaptive robot behaviour in dynamic and uncertain settings. In summary:

Contribution Statement

This thesis shows that reliable and efficient robot manipulation in unstructured environments requires motion planning and control methods that combine implicit reactivity with explicit reasoning about uncertainty.

	Chapter 3	Chapter 4	Chapter 5	Chapter 6
Theme	Reactivity	Learning for Reactivity	Robustness	Exploration
Uncertainty Handling	Implicit	Implicit	Explicit	Explicit
Prior Knowledge of Uncertainty	None	None	Sampling-based access	Sampling-based access
Environment	Deterministic	Deterministic	Stochastic	Stochastic
Method	MPC	MPC	MPC	Belief-space control
Control Problem	Min. cost	Min. cost	Min. cost s.t. chance constraints	Max. information gain

Table 1.2: Thesis overview: summary of uncertainty modelling, assumptions, and solution methods across chapters. This table is repeated at the start of each chapter.

While achieving this vision ultimately requires progress in perception, learning, and hardware, the methods and experiments presented here highlight how implicit and explicit approaches to uncertainty can be integrated to anticipate and address the challenges of real-world robotic manipulation.

1.1.1 Implicit Uncertainty Handling through Reactivity

In the spirit of implicit uncertainty handling, we develop sampling-based Model Predictive Control (MPC) methods that can replan in real-time while efficiently exploring the contact space without gradient information in Chapters 3 and 4. We note that these methods do not require prior knowledge of the uncertainty, but rather rely on high-frequency replanning to adapt to changes in the environment.

VP-STO (Chapter 3) Our first contribution is *Via-Point-based Stochastic Trajectory Optimisation* (VP-STO), a sampling-based trajectory optimisation method designed to efficiently generate *adaptive*, contact-rich manipulation behaviours in real time. VP-STO optimises over a low-dimensional via-point parameterisation of trajectories using zero-order methods to efficiently explore the contact space without gradient information. These via-points are then used to synthesise smooth

timing-optimal trajectories from optimal basis functions (Jankowski et al., 2022). This formulation enables efficient optimisation loops, probabilistic warm-starting, and effective exploration of discontinuous cost landscapes, such as those arising in contact-rich tasks. VP-STO therefore provides a practical means of producing timing-optimal, contact-making trajectories without relying on gradient information.

GPC (Chapter 4) Our second contribution is the *Generative Predictive Control* (GPC) framework, which addresses the limitations of VP-STO in more complex, high-dimensional settings where simple contact dynamics models are insufficient and accurate handling of numerous, simultaneous contacts is essential (e.g., whole-body contact-rich control). GPC amortises sampling-based Model Predictive Control (SPC) by bootstrapping it with generative models trained on SPC trajectories collected in simulation. By learning meaningful proposal distributions directly from noisy SPC data, GPC enables more efficient and informed sampling during online planning. The method relies on offline data collection with more extensive sampling and longer horizons than are feasible in an online setting, while preserving the flexibility and adaptability of online optimisation. We demonstrate GPC's effectiveness in high-dimensional loco-manipulation tasks with a quadruped robot with an arm performing non-prehensile manipulation in complex environments.

1.1.2 Explicit Uncertainty Handling through Probabilistic Reasoning

While classical MPC can provide a degree of robustness to uncertainty through its receding-horizon structure, its deterministic formulation typically renders it inherently inadequate for explicitly handling uncertainty (Mesbah, 2016). Therefore, Chapters 5 and 6 focus on alternative methods which *explicitly* handle uncertainty. While Chapter 5 addresses the question of how to ensure constraint satisfaction in stochastic environments, Chapter 6 addresses the question of how to actively reduce uncertainty in such environments through informative actions.

CC-VPSTO (Chapter 5) Approaches to constraint satisfaction in stochastic environments generally fall into two categories: *robust* methods, which enforce constraints under all possible uncertainty realizations, and *chance-constrained* methods, which require constraints to hold with high probability. Robust methods often lead to excessive conservatism and even infeasibility in practice, chance-constrained methods provide a more flexible framework for balancing safety and performance. However, existing formulations typically rely on restrictive assumptions about the uncertainty distribution (e.g., Gaussianity) or demand prohibitively large sample sizes to ensure reliable constraint satisfaction, limiting their use in real-time online control. To address this gap, our third contribution is *Chance-Constrained VP-STO* (CC-VPSTO), a Monte Carlo-based framework for online robot motion planning under uncertainty. CC-VPSTO introduces a surrogate formulation for chance-constrained trajectory optimisation that achieves low conservatism with limited samples while avoiding the overly optimistic behaviour of naïve sample average approximation methods. By integrating this surrogate into VP-STO, we extend it to a stochastic MPC framework that is compatible with real-time operation and supports arbitrary uncertainty distributions as well as general inequality constraints such as collision avoidance, force limits, and performance objectives. We provide a theoretical analysis showing that, under independence assumptions, the surrogate yields solutions that satisfy the original chance constraints with high confidence, and we argue why the approach remains effective in receding-horizon MPC settings where these assumptions do not strictly hold. Empirically, we demonstrate that CC-VPSTO consistently balances probabilistic constraint satisfaction and task efficiency across challenging simulation benchmarks and real-world robot experiments.

Yet, while CC-VPSTO provides a practical means of ensuring constraint satisfaction in stochastic environments, it does not address the question of how to actively reduce uncertainty through informative actions. This is particularly relevant in scenarios where the robot has the ability to gather information about the environment, such as through touch or other sensory modalities. In the context of contact-rich manipulation, interacting with objects can provide valuable information

about their properties, which can be used to reduce uncertainty and improve task performance.

Touch-Based Object Localisation (Chapter 6) Our final contribution is a framework for *touch-only global object localisation* that operates directly in continuous state spaces from uninformed, non-parametric priors, enabling robots to localise and manipulate objects even when vision is unreliable or absent. The approach combines a proximity-aware measurement model that turns sparse binary contact signals into informative likelihoods with a contact-aware resampling strategy that mitigates particle starvation in discontinuous observation settings. To actively reduce uncertainty, we introduce a sampling-based information-gathering controller that evaluates candidate probing actions using a non-parametric differential entropy estimator, capturing both observation-driven changes in particle weights and dynamics-driven changes in spatial density. We argue that this dual consideration is crucial for effective uncertainty reduction in scenarios where the robot directly interacts with the environment. We demonstrate that this system reliably localises and grasps objects under broad, multi-modal initial uncertainty with up to 0.4 metres of separation between modes in real-world experiments, substantially extending the applicability of touch-based localisation beyond the narrow ranges assumed in prior work.

1.2 Research Beyond This Thesis

In addition to the publications that form the basis of this thesis, I have contributed to other research projects during my DPhil. A list of these publications is provided below:

- Staniaszek, M.,Brudermüller, L., Bhattacharyya, R., Lacerda, B., and Hawes, N. (2023). Difficulty-aware time-bounded planning under uncertainty for large-scale robot missions. *European Conference on Mobile Robots (ECMR).*

- Jankowski, J., Brudermüller, L., Hawes, N., and Calinon, S. (2025). Robust pushing: Exploiting quasi-static belief dynamics and contact-informed optimization. *The International Journal of Robotics Research*.

- Staniaszek, M., Brudermüller, L., You, Y., Bhattacharyya, R., Lacerda, B., and Hawes, N. (2025). Time-bounded planning with uncertain task duration distributions. *Robotics and Autonomous Systems*.

- Ilyes, R., Brudermüller, L., Hawes, N., and Lacerda, B. (2025). Ro-To-Go! Robust Reactive Control with Signal Temporal Logic. *arXiv preprint arXiv:2503.05792*.

- Ilyes, R., Brudermüller, L., Hawes, N., and Lacerda, B. (2025). Receding Horizon Control for Signal Temporal Logic using Robustness-Conserving Partial Formula Evaluation. *Robotics and Automation Letters*.

2 Background

Contents

In this chapter, we introduce the necessary concepts and methods that are relevant to this thesis, alongside the related work in the field. We cover the

fundamentals of low-level robot control; trajectory optimisation and representation; sampling-based planning and control; uncertainty-aware planning and control; and learning for robot control. At the end of each section, we highlight how these concepts relate to the work presented in this thesis.

2.1 Robot Dynamics and Control

In this section, we provide a brief overview of the fundamentals of modelling and controlling high degree-of-freedom (DoF) robotic systems. In particular, we focus on rigid body dynamics of robot manipulators and approaches to controlling them in the presence of, and through, contact. For a more comprehensive introduction to robot dynamics and control, we refer the reader to Tedrake (2023); Lynch and Park (2017).

2.1.1 Rigid-Body Dynamics

Robot movements can be well-described by rigid-body dynamics (Featherstone, 2008) which assume that the robot's links are rigid and actuated by control torques and/or forces acting on the joints. The state of a robotic system is typically described by its generalized coordinates $\boldsymbol{q}$ (e.g., joint angles for a manipulator) and their time derivatives $\dot{\boldsymbol{q}}$ (e.g., joint velocities). The dynamics of a robotic system in standard manipulator form can be described by the following equation of motion:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{u} + \boldsymbol{\tau}_{\text{ext}}, \tag{2.1}$$

where $\boldsymbol{M}(\boldsymbol{q})$ is the mass/inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ are the Coriolis and centrifugal forces, $\boldsymbol{g}(\boldsymbol{q})$ the gravitational forces, and $\boldsymbol{\tau}_{\text{ext}}$ represents external forces and torques acting on the robot, such as contact forces. Last, the control input $\boldsymbol{u}$ represents the generalized forces generated by the robot's actuators, assuming that the robot can accurately output the desired torques/forces. This is a common assumption for high-DoF robotic manipulators, which are typically equipped with high-performance joint-level controllers that can accurately track desired torque/force commands.

While most of the experiments conducted in the works presented in this thesis are performed on robot manipulators, Chapter 4 uses a quadruped robot with an arm

as the robotic platform. While at a high level, the dynamics of a quadruped robot can also be described by Eq. (2.1), quadruped robots are typically underactuated, meaning that they have fewer actuators than degrees of freedom, e.g. the floating base of the robot is not actuated. While this influences the control strategies used for quadruped robots, the underlying dynamics and contact modelling remain the same. More specifically, the low-level control of the quadruped robot in Chapter 4 is abstracted away through a Reinforcement Learning (RL) policy that maps from higher-level commands to low-level joint torques (Zhu et al., 2025). We therefore do not discuss the specific control strategies used for quadruped robots in this thesis.

2.1.2 Manipulator Control

With the main goal of this thesis being to develop planning and control methods for contact-rich manipulation, we focus on control strategies that can be used to control robot manipulators in the presence of contact. In the following, we briefly discuss two widely used control strategies in manipulation that may serve as low-level control layer which simplifies the optimal control problem.

Direct Force Control Direct force control (Siciliano and Villani, 1999) aims to directly specify interaction forces between the robot and the environment. Therefore, the idea is to compensate for gravity and Coriolis effects, such that

$$\boldsymbol{u} = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\tau}_{\text{des}}, \tag{2.2}$$

where $\boldsymbol{\tau}_{\text{des}}$ is the new auxiliary control input that specifies the desired generalized forces acting on the robot. If the robot is in contact with the environment, the external forces balance the desired forces, leading to a quasi-static equilibrium with

$$\boldsymbol{\tau}_{\text{ext}} = -\boldsymbol{\tau}_{\text{des}}. \tag{2.3}$$

In this case, $\boldsymbol{\tau}_{\text{des}}$ directly regulates the interaction force applied to the environment. Conversely, if the robot is not in contact, i.e. $\boldsymbol{\tau}_{\text{ext}} = 0$, the commanded force produces an acceleration of the robot in the direction of $\boldsymbol{\tau}_{\text{des}}$,

$$\ddot{\boldsymbol{q}} = \boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{\tau}_{\text{des}}. \tag{2.4}$$

This dual behaviour highlights both the strength and the challenge of direct force control. While it enables accurate force regulation in sustained contact, it can lead to undesirable accelerations and unstable behaviour in free space. Furthermore, the method is highly sensitive to modelling errors and sensor noise, making it less robust for tasks that require both free-space motion and intermittent contacts. For these reasons, direct force control is often complemented with higher-level strategies that ensure stability and robustness in practical manipulation scenarios.

Stiffness Control Stiffness control, in contrast, prescribes how the robot should react to deviations in position and velocity when subject to external forces (Hogan, 1984). Rather than interpreting forces as inputs, the controller generates generalised forces $\boldsymbol{\tau}$ such that the closed-loop robot dynamics emulate a virtual spring–damper system. A standard stiffness control law is given by

$$\boldsymbol{u} = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{K}(\boldsymbol{q} - \boldsymbol{q}_{\text{des}}) - \boldsymbol{D}(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_{\text{des}}) + \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\text{des}}, \tag{2.5}$$

where $\boldsymbol{K}$ and $\boldsymbol{D}$ are the stiffness and damping matrices, and $(\boldsymbol{q}_{\text{des}}, \dot{\boldsymbol{q}}_{\text{des}}, \ddot{\boldsymbol{q}}_{\text{des}})$ define the desired motion. Substituting this into the manipulator dynamics in Eq. (2.1) leads to the closed-loop form

$$\boldsymbol{M}(\boldsymbol{q})(\ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_{\text{des}}) + \boldsymbol{D}(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_{\text{des}}) + \boldsymbol{K}(\boldsymbol{q} - \boldsymbol{q}_{\text{des}}) = \boldsymbol{\tau}_{\text{ext}}. \tag{2.6}$$

In steady-state contact ($\dot{\boldsymbol{q}} = \ddot{\boldsymbol{q}} = 0$), the external force is given by

$$\boldsymbol{\tau}_{\text{ext}} = \boldsymbol{K}(\boldsymbol{q} - \boldsymbol{q}_{\text{des}}), \tag{2.7}$$

demonstrating that the interaction force is linearly related to the position control error through the stiffness matrix $\boldsymbol{K}$. If the robot is not in contact ($\boldsymbol{\tau}_{\text{ext}} = 0$), the system converges to the desired position $\boldsymbol{q} = \boldsymbol{q}_{\text{des}}$. By tuning the virtual stiffness and damping, the robot can be made more compliant or rigid in response to external perturbations, while still having Lyapunov-stability guarantees (Albu-Schäffer et al., 2007). Unlike force control, stiffness control provides stable and predictable behaviour both in contact and free space, as the controller directly regulates the

interaction dynamics. This makes stiffness control a widely adopted strategy for contact-rich manipulation tasks where robots must simultaneously ensure accurate trajectory tracking and safe, compliant interaction with uncertain environments.

Relevance for This Thesis

We leverage these properties of stiffness control in this thesis for planning for both the robot's motion and interaction forces simultaneously through optimal control approaches that optimise for the set point of a stiffness controller.

2.1.3 Modelling Contact Dynamics

We have so far discussed the dynamics of robot manipulators, how they respond to control inputs and external forces and how different control strategies can be used to regulate the robot's motion in the presence of contact. However, we are not only interested in the robot's response to contact forces, but also in the environment's response to the robot's actions. For instance, when the robot pushes an object, how does the object move in response to the applied force? To answer this question, we require models that describe the contact dynamics between two bodies in contact. These models can then be used to predict the evolution of the robot and environment state in response to the robot's actions, which is essential for planning and control. Depending on the task and algorithmic requirements, different contact models can be used, each with its own assumptions and trade-offs. In the following, we briefly discuss two classes of contact models for rigid bodies that are relevant to this thesis. Note that we do not discuss deformable contact models, which are outside the scope of this thesis.

Second Order Models The first class of contact models are second-order models that combine Newtonian mechanics of rigid bodies with compliant contact models that consider contact forces as a function of penetration depth and relative velocity. The system state therefore includes both the generalized coordinates $\boldsymbol{q}$ and their derivatives $\dot{\boldsymbol{q}}$. Most commonly the resulting dynamics models are formulated as Linear Complementarity Problems (LCP) (Anitescu and Potra, 1997; Stewart, 2000)

that enforces non-penetration constraints and Coulomb friction within a velocity-based time-stepping scheme. This approach can accurately model rigid-body contact with friction and is widely used in physics engines such as Bullet (Coumans, 2015) and Mujoco (Todorov et al., 2012). Yet, in order to be accurate, these models require small time-steps (due to the stiff differential equations from the compliant contact models) and can thus be computationally expensive to simulate.

Quasi-Static Models The second class of contact models are quasi-static models (Mason, 1986, 2001; Pang and Tedrake, 2021). Those models assume that dynamic effects related to velocities and accelerations are negligible. However, this limits their applicability to the prediction of slow physical interactions between rigid bodies. In the context of robotics, these interactions typically involve pushing, insertion or other manipulation tasks where the robot moves slowly. This simplification leads to a significant reduction in computational complexity which comes from a reduction in the state space (only $\boldsymbol{q}$ is required) and lower temporal resolutions required for accurate simulation.

Relevance for This Thesis

We use both second-order and quasi-static contact models in this thesis. The choice of model depends on the task and algorithmic requirements. For instance, in Chapter 3, we use a quasi-static contact model for planning pushing motions in a highly dynamic environment, where fast re-planning is essential. In contrast, in Chapters 4 and 6, we use a second-order contact model to accurately simulate the dynamics of a quadruped robot and a robot manipulator, respectively, in contact-rich tasks.

2.2 Numerical Optimisation

In its most general form, a numerical optimisation problem can be expressed as

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & c(\boldsymbol{x}) \\ \text{subject to} \quad & g(\boldsymbol{x}) \leq 0, \\ & h(\boldsymbol{x}) = 0, \end{aligned} \tag{2.8}$$

where $\boldsymbol{x}$ is a vector of decision variables, c is the cost function to be minimised, and g and h are (vectors of) inequality and equality constraints, respectively. Almost any planning or control problem can be formulated as an optimisation problem of this form, but not all problems are easy to solve. Depending on the properties of the cost function and constraints, different optimisation methods can be used to find a solution $\boldsymbol{x}^*$ that minimises the cost while satisfying the constraints. These range from first-order and second-order methods that use derivative information to zero-order derivative-free methods to grid-based search methods. If c and g are convex functions and h is affine, the optimisation problem is convex, meaning that any local minimum is also a global minimum. Convex optimisation problems can be solved efficiently using well-established algorithms (Boyd and Vandenberghe, 2004). However, many real-world problems, especially in robotics, are non-convex due to complex dynamics, non-linear constraints, and discontinuities (e.g., from contact). Non-convex problems are generally more challenging to solve and may require specialised algorithms or heuristics to find good solutions. These challenges are particularly pronounced in contact-rich manipulation tasks, where the dynamics can be highly non-linear and discontinuous. Addressing such problems within fast feedback loops is a central objective of this thesis. The subsequent two sections therefore serve to establish the necessary background: first, we introduce *optimal control* as a way of formulating planning and control problems in robotics; and second, we discuss methods for solving these problems, with a particular emphasis on derivative-free approaches that are well-suited to contact-rich manipulation.

2.3 Optimal Control

Many problems in robotics are inherently *temporal*: the objective is to determine a sequence of control actions and resulting states that optimises a performance criterion over time. Optimal control provides a mathematical framework for addressing such problems by formulating the goal of control as the long-term optimisation of a scalar cost function (Bertsekas, 2012).

Consider a system with state $\boldsymbol{x}(t) \in \mathcal{R}^n$ and control input $\boldsymbol{u}(t) \in \mathcal{R}^m$ at time $t \in \mathcal{R}$, evolving according to the continuous-time dynamics

$$\dot{\boldsymbol{x}}(t) = f_c(\boldsymbol{x}(t), \boldsymbol{u}(t)), \tag{2.9}$$

where $f : \mathcal{R}^n \times \mathcal{R}^m \to \mathcal{R}^n$ represents, for example, the manipulator or contact dynamics discussed in Sec. 2.1.2. To formulate tractable optimisation problems, the dynamics are typically discretised in time, yielding

$$\boldsymbol{x}_{t+1} = f_d(\boldsymbol{x}_t, \boldsymbol{u}_t), \tag{2.10}$$

where $\boldsymbol{x}_t$ and $\boldsymbol{u}_t$ denote the state and control at discrete time step $t \in \mathbb{N}$, and f_d denotes the discrete-time transition function obtained, e.g., via numerical integration of Eq. (2.9). Unless stated otherwise, f will refer to the discrete-time dynamics in the following. In the *finite-horizon* setting[1], the desired system behaviour is encoded through the optimisation objective J, typically consisting of a terminal cost $c_T : \mathcal{R}^n \to \mathcal{R}$ that quantifies the cost of being in state $\boldsymbol{x}_T$ at the end of the time horizon T, and a running cost $c : \mathcal{R}^n \times \mathcal{R}^m \to \mathcal{R}$, $c(x_t, u_t)$ that quantifies the immediate cost of being in state $\boldsymbol{x}_t$ and applying control $\boldsymbol{u}_t$:

$$J(\boldsymbol{x}_{0:T}, \boldsymbol{u}_{0:T-1}) = c_T(\boldsymbol{x}_T) + \sum_{t=0}^{T-1} c(\boldsymbol{x}_t, \boldsymbol{u}_t). \tag{2.11}$$

[1] In infinite-horizon problems, the sum extends to infinity and a discount factor is typically introduced to ensure convergence.

2.3.1 Trajectory Optimisation

Optimising over the space of all possible control functions and state trajectories in order to solve the finite-horizon optimal control problem is generally intractable. Instead, rather than trying to solve for the optimal feedback controller for the entire state space, trajectory optimisation attempts to find an optimal control solution that is valid from only a single initial condition. In other words, instead of solving for a closed-loop policy $\pi^*(\boldsymbol{x}_t)$ that maps states to actions, i.e., $\boldsymbol{u}^* = \pi^*(\boldsymbol{x}_t)$, the goal is to find an *open-loop* trajectory $(\boldsymbol{x}^*_{0:T}, \boldsymbol{u}^*_{0:T-1})$, i.e., a sequence of states $\boldsymbol{x}^*_{0:T} = [\boldsymbol{x}^*_0, \boldsymbol{x}^*_1, \ldots, \boldsymbol{x}^*_T]$ and corresponding control actions $\boldsymbol{u}^*_{0:T-1} = \left[\boldsymbol{u}^*_0, \boldsymbol{u}^*_1, \ldots, \boldsymbol{u}^*_{T-1}\right]$ that minimises the objective J when starting from a specific initial state $\boldsymbol{x}_0$. We can still use trajectory optimisation for closed-loop control by repeatedly re-optimising the trajectory from the current state in a receding-horizon fashion, as done in Model Predictive Control (see Sec. 2.5).

There exist two main approaches to how the optimal control problem can be transcribed into a finite-dimensional nonlinear optimisation problem, that can be solved using numerical optimisation techniques (Posa and Tedrake, 2013):

1. **Direct methods** include both control and state variables as decision variables, and enforce the system dynamics as constraints, typically at a finite set of time points (collocation points). This results in a large but sparse optimisation problem that can be solved using standard numerical solvers.

2. **Shooting methods**, such as Differential Dynamic Programming (Mayne, 1973), use only the controls as decision variables by enforcing the dynamics through forward simulation. This means that the state trajectory is implicitly defined by the control trajectory and the initial state. Thus the optimiser does not need to explicitly enforce the system dynamics, resulting in a reduced search space while only considering physically-realisable trajectories. Consequently, shooting methods can employ derivative-free optimisation methods, unlike direct methods which require dynamics derivatives, as discussed below.

The two approaches lead to different formulations of the optimisation problem:

$$\textbf{Direct:} \quad \min_{\boldsymbol{u}_{0:T-1}, \boldsymbol{x}_{0:T}} \; J(\boldsymbol{x}_{0:T}, \boldsymbol{u}_{0:T-1}) \tag{2.12a}$$

$$\textbf{Shooting:} \quad \min_{\boldsymbol{u}_{0:T-1}} \; J(\boldsymbol{x}_{0:T}, \boldsymbol{u}_{0:T-1}) \tag{2.12b}$$

$$\text{subject to} \quad \boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t), \tag{2.12c}$$

$$\boldsymbol{x}_0 = \boldsymbol{x}_{\text{init}}, \tag{2.12d}$$

$$\text{other constraints (e.g., state/control bounds).} \tag{2.12e}$$

Finally, we note that hybrid approaches also exist. For instance, multiple shooting (Giftthaler et al., 2018), which combines aspects of both direct and shooting methods by dividing the time horizon into segments and optimising over both states and controls at the segment boundaries while enforcing dynamics within each segment through simulation. Yet, multiple shooting methods still require dynamics derivatives and are therefore not discussed further in this thesis.

Which transcription is more suitable for contact-rich manipulation? Both transcription methods have been successfully applied to formulate optimal control problems for contact-rich manipulation. Direct methods include contact-implicit formulations (Posa and Tedrake, 2013; Manchester and Kuindersma, 2019), where discrete contact modes are implicitly incorporated through a complementary contact constraint, which enforces the interaction force *or* the shortest distance between two bodies to be zero. This enables reasoning over making and breaking contacts while optimising state and action trajectories simultaneously. On the other hand, shooting methods approach the problem by directly simulating contact dynamics within the optimisation loop, as in iLQR-based approaches (Tassa et al., 2012) or sampling-based methods, such as Predictive Sampling (Howell et al., 2022a) and Model Predictive Path Integral (MPPI) control (Bhardwaj et al., 2022; Williams et al., 2017).

Relevance for This Thesis

In this thesis, we focus on shooting methods for trajectory optimisation in contact-rich manipulation tasks with system dynamics enforced through forward simulation via the contact models introduced earlier. The main reason for this choice is that shooting methods allow for more flexible and lower-dimensional trajectory representations, which can significantly reduce the computational complexity of the optimisation problem. This is particularly important in contact-rich scenarios, where the dynamics can be highly non-linear and discontinuous, making the optimisation problem more challenging to solve. Furthermore, shooting methods can leverage derivative-free optimisation techniques, which are often more robust to the non-smoothness and discontinuities inherent in contact dynamics. This makes them well-suited for real-time applications, where fast re-planning is essential. We discuss these aspects in more detail below.

2.3.2 Trajectory Representation

A key aspect of trajectory optimisation is the choice of trajectory representation, which can significantly affect the efficiency and effectiveness of the optimisation process, particularly in sampling-based methods. Lower-dimensional representations entail reduced search spaces, which can lead to faster convergence and reduced computational costs. The choice of representation depends on the specific requirements of the task, such as the need for smoothness, continuity, or the ability to represent complex motions, as well as the transcription method used.

Fixed Time Discretisation Direct transcription methods typically require a fixed time discretisation of the trajectory, meaning that the state and control inputs are represented at fixed discrete time steps. The respective resolution is typically linked to the discretised system dynamics in Eq. (2.10). We discussed above how contact modelling approaches can influence the required time resolution for accurate simulation. Yet, higher temporal resolutions lead to higher-dimensional optimisation

problems, which can be challenging to solve, especially in real-time applications. The trade-off between simulation accuracy and problem dimensionality is a key consideration when using fixed time discretisations for trajectory representation in contact-rich manipulation tasks.

Lower-Dimensional Representations Discretising a trajectory in time provides a straightforward way to make trajectory optimisation problems computationally tractable. However, this approach couples the number of decision variables directly to the temporal resolution at which the system dynamics are enforced, which can make the optimisation problem unnecessarily high-dimensional. A common approach to reduce the dimensionality of the trajectory representation is to use a superposition of basis functions, such as B-splines (De Boor and De Boor, 1978), Radial Basis Functions (RBFs) Powell (1987), or Optimal Basis Functions (OBFs) (Jankowski et al., 2022). The resulting trajectory can therefore be expressed as

$$\boldsymbol{q}(t) = \sum_{n=1}^{N} \boldsymbol{\phi}_n(t) w_n, \tag{2.13}$$

where $\boldsymbol{\phi}_n(t)$ denotes the n-th basis function and w_n its associated weight. The weights serve as the decision variables in the optimisation problem, so the dimensionality of the optimisation is determined directly by the number of basis functions N. Because the trajectory is expressed as a linear combination of continuous functions, it remains continuous in time, and constraints can be evaluated at arbitrary temporal resolutions. This approach allows for a more compact representation of the trajectory, as the number of decision variables is decoupled from the temporal resolution, at which the system dynamics are enforced. Yet, this representation comes with a trade-off: the expressiveness of the trajectory is restricted to the span of the chosen basis functions. For underactuated systems such as robots interacting with objects, this limitation prevents the direct use of basis function superpositions to represent the full system trajectory. Nevertheless, the benefits of decoupling decision variables from temporal resolution can still be exploited by applying shooting methods to

optimise the system's control inputs, as introduced in Section 2.3.1. In this setting, the control trajectory is parameterised by basis functions, while the system dynamics are used to generate the corresponding state trajectory. This enforces dynamic consistency without placing additional constraints on the control trajectory itself. Eq. (2.13) can also be used to define the set point of a low-level stiffness controller, as discussed in Section 2.1.2, while the robot's closed-loop motion and force response are abstracted away through the stiffness controller.

Relevance for This Thesis

Given the focus on shooting methods in this thesis, we exploit lower-dimensional trajectory representations based on basis functions to improve computational efficiency in contact-rich manipulation tasks. This parameterisation decouples the number of decision variables from the temporal resolution at which the system dynamics are enforced, enabling compact, continuous-time trajectories and flexible time-scaling without increasing the dimensionality of the optimisation problem.

2.4 Trajectory Optimisation Methods

Methods for numerical optimisation, as introduced in Section 2.2, can be broadly categorised into derivative-based (higher-order) and derivative-free (zero-order) methods. In the following, we provide a brief overview of both classes of methods, with a particular focus on their applicability to contact-rich manipulation tasks and the respective transcription methods introduced above.

2.4.1 Derivative-Based (Higher-Order) Optimisation

Derivative-based, methods have been widely used for trajectory optimisation in robotics, e.g., (Tassa et al., 2012; Le Cleac'h et al., 2024; Posa and Tedrake, 2013). These methods typically rely on the availability of accurate gradients of the dynamics and cost function, which can be challenging to obtain in contact-rich scenarios due to discontinuities and non-smoothness in the dynamics and cost functions.

Smoothing of contact dynamics alone – as done in differentiable simulators, such as *Dojo* (Howell et al., 2022b) – is often not sufficient to make derivative-based methods work reliably in contact-rich manipulation. These methods strongly rely on good initial guesses and are prone to getting stuck in poor local minima, especially when the problem requires mode switches, e.g. making and breaking contact. This limitation was recently emphasised by Zhang et al. (2025), who combine iLQR (Li and Todorov, 2004) with MuJoCo dynamics and finite-difference approximated derivatives for whole-body control of legged robots. While this approach performs well when a contact mode schedule is provided, typically encoded in the cost function for locomotion, it struggles to find good solutions when the mode sequence is not known a priori, as is common in more complex contact-rich manipulation tasks. Hogan and Rodriguez (2020) address this limitation in derivative-based methods by learning a mode switcher for planar pushing tasks. Yet, besides having to train an additional model, this approach is limited to the specific modes considered during training. Derivative-based methods are technically compatible with both direct and shooting transcription methods. In direct methods, e.g., (Posa and Tedrake, 2013), the system dynamics act as constraints that link state and control variables. In shooting-based formulations, the state cost is propagated back to the control variables through the dynamics, e.g., (Tassa et al., 2012).

2.4.2 Derivative-Free (Zero-Order) Optimisation

Derivative-free optimisation methods, also known as *zero-order* methods, address some of the limitations of derivative-based methods by not requiring gradient information. Instead, they rely solely on function evaluations to guide the search for optimal solutions.

Deterministic Derivative-Free Optimisation. First, although our focus in this thesis is on *stochastic*, derivative-free approaches, it is important to acknowledge the complementary class of *deterministic* derivative-free methods. These methods construct local models of the objective function and update them deterministically

using carefully designed sampling patterns. Classical examples include trust-region and interpolation-based algorithms, thoroughly reviewed by Conn et al. (2009). A widely used representative method is COBYLA (Powell, 1994), which models the objective and constraint functions via linear interpolation and solves a sequence of constrained subproblems. While powerful in smooth, lower-dimensional settings, these methods are typically less suited to the highly non-smooth systems addressed in this thesis, motivating our emphasis on stochastic zero-order techniques.

Stochastic Derivative-Free Optimisation. In contrast, the central idea of *stochastic* or *black-box* optimisation (Audet and Kokkolaras, 2016) methods is to maintain a sampling distribution over candidate solutions (e.g., trajectories or control sequences), and iteratively refine this distribution using objective evaluations. By avoiding the need for gradient information, zero-order optimisation naturally handles discontinuities and enables exploration beyond local minima. An additional advantage is that candidate evaluations are trivially parallelisable, making these methods well-suited for online settings such as model predictive control. Popular algorithms include random search (Matyas et al., 1965), evolutionary strategies, such as CMA-ES (Hansen and Ostermeier, 2001) and the Cross-Entropy Method (CEM) (Rubinstein, 1999), and genetic algorithms (Holland, 1992). A more comprehensive overview in the context of robotics is given by Jordana et al. (2025). We will provide a brief overview of some of these methods in the context of sampling-based MPC in Section 2.5 below.

As discussed in Section 2.3.1, shooting methods provide a trajectory optimisation formulation that naturally accommodates derivative-free techniques. In contrast, direct methods require the system dynamics to be enforced as constraints, which cannot be easily handled by zero-order methods. This is due to the fact that valid candidate action-state trajectories are constrained through the dynamics and thus difficult to sample. However, shooting-based formulations only require sampling action trajectories to subsequently roll-out the state trajectories. We rely on this procedure in this thesis. Thus, in the context of this work, when we refer to

stochastic/sampling-based optimisation, we imply that the underlying transcription is based on shooting methods, as in Eq. (2.12b).

2.5 Model Predictive Control (MPC)

Model Predictive Control (MPC), also known as receding-horizon control, solves an open-loop optimal control problem repeatedly over a finite time horizon H, typically much shorter than the total horizon T (Mayne et al., 2000). At each time step, given the current measured state of the system, MPC optimises the control inputs over the horizon H, applies the first control input, and then repeats the process at the next time step. In this way, MPC replaces an explicit feedback law with the repeated solution of an optimisation problem, thereby acting as an *implicit* feedback controller that can compensate for model inaccuracies and external disturbances (Mesbah, 2016). This recursive formulation makes MPC effectively a *locally optimal* state-feedback control scheme, where control inputs are continually adapted online to the observed state and predicted system evolution. MPC methods traditionally solve the problem in Eq. (2.12) over the receding horizon H using derivative-based non-convex optimisation methods, such as Sequential Quadratic Programming (SQP) (Wensing et al., 2023) or Differential Dynamic Programming (DDP) (Farshidian et al., 2017). Yet, as discussed above, these methods can struggle in contact-rich scenarios due to the non-smooth and discontinuous nature of the dynamics and cost functions.

2.5.1 Sampling-Based MPC

In response to these challenges, sampling-based MPC, also referred to as sampling-based predictive control (SPC), has gained significant attention as a simple and computationally efficient – if parallelised – alternative to derivative-based methods for MPC in robotics (Alvarez-Padilla et al., 2025; Howell et al., 2022a; Williams et al., 2017; Bhardwaj et al., 2022; Xue et al., 2025; Pezzato et al., 2025). Instead of relying on specific problem structure or derivative information, these approaches use sampling-based optimisation methods (cf. Sec. 2.4.2) to explore the control space using finite samples, either to approximate gradients, or to directly search

Algorithm 1: Sampling-Based Predictive Control

Input: initial policy parameters θ, number of samples N, cost function J, state estimator $\hat{\boldsymbol{x}}(\tau)$, planning horizon T

$\tau \leftarrow 0$ *// Initialize time step*

while *planning* **do**

  $\boldsymbol{x}_\tau \leftarrow \hat{\boldsymbol{x}}(\tau)$ *// Get current state estimate*

  for $i = 1$ **to** N ***in parallel*** **do**

    Sample Controls $U_\tau^{(i)} \sim \pi_\theta(U)$

    Compute $J^{(i)} \leftarrow J(U_\tau^{(i)}; \boldsymbol{x}_\tau)$ *// via forward simulation*

  $\theta \leftarrow \texttt{update\_params}(\{U_\tau^{(i)}, J^{(i)}\}_{i=1}^N)$

  $\boldsymbol{u}_\tau \leftarrow \texttt{get\_action}(\theta, \tau)$ *// e.g., via spline interpolation*

  $\texttt{execute}(\boldsymbol{u}_\tau)$

  $\tau \leftarrow \tau + 1$

over candidate control sequences, and solve the optimal control problem in a receding-horizon fashion.

For simplicity, in the following, we rewrite the shooting objective in Eq. (2.12b) in a more compact form by denoting the H-length control sequence as $U_\tau = [\boldsymbol{u}_\tau, \boldsymbol{u}_{\tau+1}, \ldots, \boldsymbol{u}_{\tau+H-1}]$, i.e.,

$$\min_{U_\tau} \; J(U_\tau; \boldsymbol{x}_\tau), \tag{2.14}$$

where $\boldsymbol{x}_\tau$ is the current state at time step τ. Again, note that this highly non-convex objective comprises the cost, as well as the system dynamics, which are enforced through forward simulation. A general outline of a general SPC algorithm is given in Algorithm 1. At each control step, SPC algorithms draw N candidate sequences $\{U_\tau^{(i)}\}_{i=1}^N$ from a parameterised sampling distribution π_θ. Each candidate is rolled out to evaluate its cost $J^{(i)} = J(U_\tau^{(i)}; \boldsymbol{x}_\tau)$ via forward simulation of the system dynamics. The distribution parameters θ are updated based on the evaluated samples $\{U_\tau^{(i)}, J^{(i)}\}_{i=1}^N$, with the precise update rule determined by the chosen sampling-based optimisation method. The first control of the refined distribution is then applied, and the process is repeated at the next time step. The update step typically involves re-fitting the sampling distribution to the best-performing samples, thereby biasing future samples towards more promising regions of the solution space. Examples include exponential cost-weighting as in MPPI (Williams et al., 2017),

or elite-set refitting as in the Cross-Entropy Method (CEM) (Rubinstein, 1999). Although SPC is not restricted to any particular distribution family, Gaussian distributions are most commonly employed due to their simplicity, tractable parameterisation, and efficient sampling, i.e.,

$$U_\tau^{(i)} \sim \mathcal{N}(\bar{U}_{\tau-1}, \Sigma_{\tau-1}), \quad i \in [1, N], \tag{2.15}$$

where the general update rule of $\bar{U}_\tau$, according to some weighting function $g : \mathcal{R} \to \mathcal{R}^+$, is given by

$$\bar{U}_\tau = \bar{U}_{\tau-1} + \frac{\sum_{i=1}^N g(J^{(i)}) \left(U^{(i)} - \bar{U}_{\tau-1}\right)}{\sum_{i=1}^N g(J^{(i)})}, \tag{2.16}$$

as summarised by Kurtz and Burdick (2025). Typical choices for the weighting function $g(J)$ include:

- *Model Predictive Path Integral (MPPI)* (Williams et al., 2017):

  $$g_{\text{MPPI}}(J) = \exp(-J/\lambda)\,,$$

  where $\lambda > 0$ is a temperature parameter. Smaller λ places more weight on low-cost samples.

- *Predictive Sampling (PS)* (Howell et al., 2022a):

  $$g_{\text{PS}}(J) = \lim_{\lambda \to 0} \exp(J/\lambda)\,,$$

  which corresponds to selecting the single lowest-cost sample.

- *Cross-Entropy Method (CEM)* (Rubinstein, 1999):

  $$g_{\text{CEM}}(J) = \begin{cases} 1 & \text{if } J \leq \gamma, \\ 0 & \text{otherwise,} \end{cases}$$

  where γ is a threshold implicitly defined by a fixed number of elite samples.

All of the above update rules have in common that the sampling distribution is derived through an exponential transformation of the cost function, which can be interpreted as a likelihood function in a probabilistic inference framework, i.e.,

$$p(U_\tau) \propto \exp\left(-J(U_\tau)\right). \tag{2.17}$$

While traditional implementations of MPPI and Predictive Sampling only update the mean of the Gaussian proposal distribution while maintaining a fixed, typically diagonal, covariance matrix, CEM does adapt the covariance—though in practice it is often restricted to a diagonal form. More sophisticated algorithms such as CMA-ES (Hansen and Ostermeier, 2001) perform full covariance adaptation, enabling richer exploration of the solution space. In addition, variants like Tsallis-MPPI (Wang et al., 2021b) and Dial-MPC (Xue et al., 2025) also extend MPPI by incorporating covariance adaptation or multi-step updates, respectively.

Practical Enhancements Several algorithmic refinements are commonly employed to make sampling-based MPC effective in practice. First, optimisation is often *warm-started* using the previous solution, since successive control problems differ only slightly; this bootstrapping scheme uses θ_{t-1} as the initialisation for the next round of optimisation. Second, relatively *short (myopic) horizons* can already be sufficient to induce the desired long-term behaviour, reducing the computational burden of forward simulations without sacrificing performance. Finally, it is rarely necessary to solve the optimisation problem to full convergence; instead, *approximate solutions* obtained within a limited number of iterations are typically adequate for achieving robust closed-loop control.

Relevance for This Thesis

This thesis focuses on the use of *sampling-based* methods for trajectory optimisation, in particular in the context of MPC, due to their robustness to non-smooth dynamics and cost functions commonly encountered in robotics. By leveraging the strengths of these methods, we aim to improve the efficiency and effectiveness of model-based planning and control in robot manipulation tasks.

2.6 Planning and Control Under Uncertainty

Up to this point, we have discussed planning and control methods that assume perfect knowledge of the system dynamics and environment. While some methods can handle model inaccuracies and disturbances through reactive feedback control,

they typically assume that the underlying system dynamics are deterministic and known. However, this assumption may not hold in practice due to various sources of *uncertainty*, such as sensor noise, model inaccuracies, and unpredictable environmental interactions. These uncertainties can significantly affect the performance and safety of robotic systems if not properly accounted for. Therefore, explicitly reasoning about these uncertainties, beyond reactive feedback control, is crucial for robust and reliable robot behaviour in real-world applications.

In the following, we focus on two main types of uncertainty: *stochasticity* in the system dynamics and *partial observability* of the system state. We start by discussing stochastic models and how to incorporate them into optimal control problems in open-loop and closed-loop settings. We then extend this discussion to partially observable systems, where we do not have direct access to the true system state, but instead receive observations that have a probabilistic relationship to the underlying state.

2.6.1 Stochastic Models

In Section 2.3.1, we introduced the discrete-time system dynamics in Eq. (2.10) as a deterministic mapping from the current state $\boldsymbol{x}_t$ and control input $\boldsymbol{u}_t$ to the next state $\boldsymbol{x}_{t+1}$. We can extend this formulation to account for stochasticity in the system dynamics by introducing a random variable $\boldsymbol{w}_t$, which captures the uncertainty in the system dynamics. The stochastic dynamics can then be expressed as

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{w}_t). \tag{2.18}$$

We distinguish between two main types of uncertainty models (Rawlings et al., 2020):

1. **Bounded uncertainty**, where the random variable $\boldsymbol{w}_t$ is assumed to lie within a known bounded set, e.g., $\boldsymbol{w}_t \in \mathcal{W}$.

2. **Probabilistic uncertainty**, where $\boldsymbol{w}_t$ is modelled as a random variable with a known probability distribution, i.e., $\boldsymbol{w}_t \sim p(\boldsymbol{w}_t \mid \boldsymbol{x}_t, \boldsymbol{u}_t)$.

2.6.2 Cost and Constraints for Stochastic Systems

Given a stochastic model, we can modify the optimal control problem in Eq. (2.12) to account for uncertainty in the system dynamics. However, this raises the question of how to define the cost function and constraints in the presence of uncertainty.

From Deterministic Costs to Risk Metrics With the predicted state being a random variable, the cost J in Eq. (2.12) also becomes a random variable. Therefore, we need to define a suitable objective that captures the desired performance under uncertainty. The objective can be defined in terms of a *risk-metric*, i.e., mapping from the cost distribution to a scalar value that quantifies the risk associated with that random variable (Majumdar and Pavone, 2017; Emmer et al., 2013). Examples of risk metrics include *expected/average cost*, measuring the average (or some time-discounted variant) performance over all possible outcomes, and *worst-case cost*, quantifying the maximum possible cost over all possible outcomes. Other risk metrics from finance and economics, looking at the tail behaviour of the distribution, include *Value at Risk (VaR)* and *Conditional Value at Risk (CVaR)*. For a given probability level $\alpha \in (0, 1)$, VaR_α is defined as the $(1-\alpha)$-quantile of the cost distribution, i.e., the cost that is not exceeded with probability α:

$$\mathrm{VaR}_\alpha(Z) := \min\{z \mid \Pr(Z \leq z) \geq \alpha\}, \tag{2.19}$$

where Z is the cost random variable. In contrast, CVaR_α is defined as the expected cost/safety over the tail of the distribution beyond the VaR_α level:

$$\mathrm{CVaR}_\alpha(Z) := \mathcal{E}[Z \mid Z \geq \mathrm{VaR}_\alpha(Z)]. \tag{2.20}$$

Figure 2.1 illustrates these commonly used risk metrics.

Constraints for Stochastic Systems In addition to defining a suitable cost function, we also need to consider generalising deterministic constraints to stochastic settings. In the deterministic case, constraints must be satisfied at all times, e.g., concerning state and control bounds, or obstacle avoidance. Especially in the case

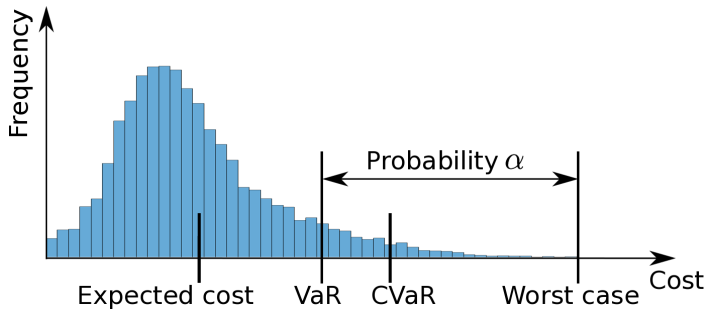


Figure 2.1: Commonly used ***risk metrics***, as illustrated by (Majumdar and Pavone, 2017): *(i)* expected cost, *(ii)* worst case, *(iii)* Value at Risk (VaR) and *(iv)* Conditional Value at Risk (CVaR).

of unbounded uncertainty, it is typically not possible to find solutions that satisfy these hard constraints. Instead, we need to "soften" the constraints to allow for some level of violation, while still providing meaningful guarantees on the system's behaviour (Rawlings et al., 2020). The risk metrics introduced above can equally be used to define such soft constraints. Since $\boldsymbol{x}$ is a random variable, the outcome of the constraint function $g(\boldsymbol{x})$ is also a random variable, and constraints can therefore be expressed using risk metrics applied to the outcomes of $g(\boldsymbol{x})$. For example, we can define *chance constraints* that require the probability of constraint violation to be below a certain threshold, i.e.,

$$\Pr\Big(g(x) > 0\Big) \leq \alpha, \tag{2.21}$$

where $\alpha \in (0, 1)$ is the acceptable probability of violation, often referred to as the risk level (Mesbah, 2016). We note that the chance constraint in Eq. (2.21) is equivalent to imposing that the α-VaR of the random variable $g(\boldsymbol{x})$ is non-positive:

$$\mathrm{VaR}_\alpha\Big(g(\boldsymbol{x})\Big) \;\leq\; 0. \tag{2.22}$$

We distinguish between *joint chance constraints*, which bound the probability of constraint violations over the entire planning horizon, and *marginal (pointwise-in-time) chance constraints*, which only bound the violation probability at each individual time step. The latter do not provide trajectory-level guarantees, and transpositions between the two are generally conservative.

2.6.3 Robust and Stochastic Control

The two uncertainty models introduced above (cf. Section 2.6.1), bounded and probabilistic uncertainty, lead to two principal approaches for planning and control of stochastic systems: *robust control* (Zhou and Doyle, 1998) and *stochastic control* (Bertsekas and Shreve, 1996). As a form of *robust optimisation* (Ben-Tal and Nemirovski, 1998), robust control seeks to guarantee constraint satisfaction under the *worst-case* realisation of bounded uncertainty. By only specifying bounds on the uncertainty set, rather than its full probability distribution, these methods are often more tractable than probabilistic formulations and connect naturally to reachability analysis (Bajcsy et al., 2019). However, worst-case reasoning can lead to overly conservative behaviour or even infeasibility (Trevisan et al., 2025), since guarantees must hold for extreme, low-probability events. In contrast, stochastic control methods (Bertsekas and Shreve, 1996) model uncertainty probabilistically and optimise the expected cost, or a more general risk measure (cf. Section 2.6.2), while ensuring probabilistic constraint satisfaction through chance constraints. This enables more flexible trade-offs between performance and robustness, tailored to the desired risk tolerance, thereby reducing conservatism and often yielding more cost-efficient behaviour (Yin et al., 2025). The main drawbacks are the need for accurate probabilistic models and the high computational cost associated with propagating uncertainty through nonlinear dynamics and reformulating chance constraints, which often involve intractable integrals (Mesbah, 2016). Two main strategies exist: *analytic reformulations*, efficient but limited to linear–Gaussian settings, and *sampling-based methods*, more general but computationally intensive (Mesbah et al., 2014). Sampling-based approaches include *scenario optimisation* (Calafiore and Campi, 2006), which approximates chance constraints using a finite number of sampled uncertainty realisations, and *Monte Carlo (MC) approximations* (Blackmore et al., 2011), which estimate the probability of constraint satisfaction through random sampling. Other stochastic control approaches replace chance constraints with other risk constraints, most commonly CVaR (Trevisan et al., 2025; Lew et al., 2025, 2023; Yin et al., 2022). While CVaR constraints enable accounting for tail

events and facilitate computational tractability due to their convexity properties, as shown by Lew et al. (2023), they can also lead to overly conservative solutions if the risk level is not chosen appropriately.

Extending these ideas to receding-horizon settings leads to *Robust MPC (RMPC)* and *Stochastic MPC (SMPC)* (Rawlings et al., 2020). Incorporating uncertainty into MPC enables systematic performance–robustness trade-offs in feedback loops, which is essential for many robotics applications, such as navigation in dynamic environments (Lew et al., 2025; Trevisan et al., 2025; de Groot et al., 2025; Hu and Fisac, 2022; Belvedere et al., 2025) and contact-rich manipulation (Jankowski et al., 2025a; Arruda et al., 2017; Agboh and Dogar, 2018). Yet, despite their potential, robust and stochastic MPC methods face significant challenges in terms of computational complexity and real-time applicability, especially for high-dimensional systems with complex dynamics. While simplifications, such as linearising the dynamics (Blackmore et al., 2011) or assuming Gaussian uncertainties (Mohamed et al., 2025), help making these problems more tractable and thus computationally efficient, they can also lead to suboptimal performance when the assumptions do not hold. While more advanced methods provide stronger theoretical guarantees, their practical applicability is often limited by computational requirements that can surpass those of standard non-robust MPC by several orders of magnitude (Belvedere et al., 2025). (Belvedere et al., 2025).

Relevance for This Thesis

In this thesis, we adopt a probabilistic treatment of uncertainty to better capture the stochastic nature of real-world contact dynamics. In contact-rich manipulation, uncertainties in object geometry, frictional properties, and contact interactions are typically *unbounded* and difficult to express through fixed bounds or worst-case assumptions. Probabilistic models offer a more faithful representation of these uncertainties, e.g., by learning stochastic dynamics or noise distributions from data (Bianchini et al., 2023), while providing a principled way to balance performance and robustness through

risk-sensitive objectives and constraints. At the same time, we aim to retain the benefits of fast feedback inherent to MPC-like schemes, ensuring that the resulting methods remain reactive and computationally tractable for real-time control. Therefore, the focus in this thesis is on developing planning and control methods that explicitly incorporate *probabilistic uncertainty models* into stochastic model predictive control formulations, enabling robust and yet adaptive behaviour in contact-rich manipulation tasks. The above discussion forms the foundation of Chapter 5.

2.6.4 Belief Space Planning

While planning and control methods often assume the current system state is *fully observable* or at least *accessible* (meaning the controller can use the true state or a reliable estimate), many real-world scenarios involve *partial observability* due to sensor noise or occlusions. Instead of relying on possibly unreliable state estimates, it is often beneficial to explicitly model this uncertainty. This is done by maintaining a *belief state*, i.e., a probability distribution over all possible system states. Planning and control methods that explicitly manage this uncertainty operate in *belief space*, the space of all possible belief states. In this framework, actions have a dual effect: they not only change the underlying world state but can also be strategically chosen to *reduce uncertainty*, e.g., by performing an *information-gathering* action (perceptual or physical). However, planning in belief space is inherently challenging for several reasons. Even coarse, finite-dimensional approximations of belief distributions lead to planning problems in spaces far higher-dimensional than the original state space. Moreover, the induced belief dynamics are typically nonlinear, underactuated (the number of control inputs is smaller than the dimension of the belief space), and stochastic, as future belief transitions depend on observations that have not yet been made.

POMDPs and Belief-MDPs This problem can be formalised as a Partially Observable Markov Decision Process (POMDP) (Kaelbling et al., 1998) within

the broader domain of sequential decision-making under uncertainty. A POMDP extends the classic *Markov Decision Process (MDP)* (Puterman, 2014) by accounting for the fact that the agent receives only noisy observations that provide incomplete information about the true system state. Critically, a POMDP can be transformed into an equivalent, but continuous-state, fully observable MDP known as a *Belief-MDP* (Åström, 1965). The states of the Belief-MDP are the belief states. The transition dynamics are defined by a *belief update* function, which takes the current belief b, the chosen action a, and the resulting observation o, and computes the next belief b'. The solution to the Belief-MDP is a *policy* $\pi(b)$ that maps a belief state to an optimal action, maximising the expected cumulative reward over time (or equivalently minimising costs).

Belief Space Control Computing the complete, optimal policy for a POMDP/Belief-MDP offline is *PSPACE-hard* (Papadimitriou and Tsitsiklis, 1987) and therefore computationally prohibitive, particularly for continuous state and action spaces. Even point-based approximations (Sunberg and Kochenderfer, 2018) that typically discretise the belief space often become intractable over long horizons. A more practical and tractable approach for real-world applications is *belief space control*. Instead of solving for the complete global policy, the goal is to find a locally optimal finite-horizon sequence of actions (a trajectory) that reaches a specific belief state (or region of the belief space) in continuous belief space given the current belief state. Casting the partially observable control problem as a fully observable, underactuated, stochastic control problem in belief space thus allows for standard planning and control techniques (Platt et al., 2010) with online replanning. We can formulate this problem as follows: let the system at time step t be described by a continuous state variable $\boldsymbol{x}_t$, which evolves according to stochastic dynamics, then

$$\boldsymbol{x}_{t+1} \sim p(\cdot \mid \boldsymbol{x}_t, \boldsymbol{u}_t), \tag{2.23}$$

where $\boldsymbol{u}_t$ denotes the continuous control action. At each time step, the system receives a measurement

$$\boldsymbol{z}_t \sim p(\cdot \mid \boldsymbol{x}_t), \tag{2.24}$$

which depends probabilistically on the true (and unknown) state of the system. Due to partial observability, the true state $\boldsymbol{x}_t$ is treated as a random variable whose distribution is represented by the *belief*

$$b_t = p(\boldsymbol{x}_t \mid \boldsymbol{u}_{0:t-1}, \boldsymbol{z}_{1:t}, b_0), \tag{2.25}$$

where b_0 is the prior belief over the initial state. The belief evolves over time through the Bayesian filtering process that combines the system dynamics and observation models. Planning and control in belief space can then be formulated as an optimal control problem over a finite horizon H:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & J(b_{1:H}, \boldsymbol{u}_{0:H-1}(\boldsymbol{\theta})) \\ \text{s.t.} \quad & b_t = p(\boldsymbol{x}_t \mid \boldsymbol{u}_{0:t-1}, \boldsymbol{z}_{1:t}, b_0), \end{aligned} \tag{2.26}$$

where $\boldsymbol{\theta}$ parameterises the control sequence $\boldsymbol{u}_{0:H-1}$, and the cost $J(\cdot)$ depends on the anticipated sequence of belief states. This optimisation problem thus seeks a control strategy that drives the belief toward a desired goal region in belief space, typically corresponding to both achieving task objectives and reducing uncertainty. Previous work has successfully established the value of control in belief space based on simplified models (e.g., assuming Gaussian belief distributions) and replanning (Platt et al., 2011; Kaelbling and Lozano-Pérez, 2013; Erez and Smart, 2012; Du Toit and Burdick, 2010; Hauser, 2011). Yet, the success and tractability of this approach still heavily depend on the belief model and representation used and the resulting cost-landscape in belief space. In the case of complex, non-Gaussian beliefs, the optimisation problem in Eq. (2.26) can become highly non-convex and difficult to solve. Also, sparse or uninformative cost functions can lead to poor local minima, as the planner may struggle to find informative actions that reduce uncertainty and improve task performance. In the latter case, it can be beneficial to augment the cost function with additional terms that explicitly encourage uncertainty reduction (Fischer and Tas, 2020; Curtis et al., 2022).

Implications for Contact-Rich Manipulation Contact-rich manipulation represents a particularly challenging instance of partially observable problems. In such settings, the robot typically has only limited information about the object (e.g., pose, shape, and mass distribution) and about the environment (e.g., contact geometry, frictional properties, or obstacles). Moreover, contact interactions themselves are inherently uncertain. While general belief space control methods aim to reduce uncertainty through observations, in contact-rich settings the dynamics themselves can be exploited for information gathering, e.g., by using funnelling push strategies that disambiguate object pose (Erdmann and Mason, 2002). However, these contact dynamics violate many of the assumptions underpinning existing belief space control methods, which often rely on linear dynamics and Gaussian uncertainty models (Platt et al., 2010; Majumdar and Tedrake, 2017). As a result, applying standard belief space planners to manipulation tasks can lead to poor approximations or intractable computations. Several works have sought to make belief space planning for manipulation tractable through simplifying assumptions, such as discretising the state and action spaces (Horowitz and Burdick, 2013; Koval et al., 2016). More recently, Marques et al. (2025) proposed a POMDP formulation for "manipulation-enhanced mapping," where manipulation actions are used to iteratively refine a belief over a spatial map. While this work introduces interesting ideas on learning belief dynamics, it remains limited to discrete state representations and does not address the challenges posed by continuous, non-smooth contact dynamics.

Relevance for This Thesis

The concepts introduced in this section form the basis for Chapter 6, which explores belief space planning for touch-based object localisation in contact-rich manipulation. The belief space perspective allows us to explicitly reason about uncertainty in the object's pose, while an information-theoretic cost encourages the robot to gather informative tactile feedback in order to maximise the probability of grasp success. Instead of relying on Gaussian approximations, we represent the belief using a particle filter, which can capture complex, multi-

modal distributions that arise naturally in contact-rich settings. Computing an information-theoretic measure on a non-parametric particle-based belief representation is non-trivial and requires careful consideration of not only the observations, but also the stochastic contact dynamics.

2.7 Learning for Robot Control

Machine learning has become an increasingly important component of robot control, providing ways to acquire policies from data and to improve the efficiency of optimisation-based approaches. Approaches to learning for robot control fall into a spectrum from direct policy acquisition via supervised or reinforcement learning to hybrid approaches that integrate data-driven models and cost functions into classical planning and control. We discuss both perspectives in the following, alongside related works of learning for contact-rich manipulation in both contexts. Finally, we review recent advances in generative modelling, which has shown great promise for both direct policy learning and model-based control.

2.7.1 Policy Learning

Learning control policies directly from data has become an increasingly effective strategy for contact-rich robotic manipulation, where modelling complex dynamics and contact transitions can be challenging. Instead of relying on sufficiently accurate analytical models, policy learning leverages data to learn mappings from observations to actions that achieve the desired goal for a given task. Among data-driven approaches, behaviour cloning and reinforcement learning represent two complementary paradigms: the former learns from expert demonstrations, while the latter learns autonomously through environmental interaction. We also note that these approaches can be combined, e.g., by using behaviour cloning to initialise a policy that is then further refined through reinforcement learning or by defining an imitation learning objective as an auxiliary reward during reinforcement learning.

Behaviour Cloning A canonical example for policy learning is *behaviour cloning* (BC), where observation-action pairs from expert demonstrations are used to learn a control policy via supervised learning (Bain and Sammut, 1995). BC methods can be broadly divided into *explicit* and *implicit* approaches. In *explicit behaviour cloning*, a function approximator (often a neural network) is trained to directly map observations or states to expert actions via supervised learning. In the case of full-state feedback, this corresponds to learning a deterministic mapping $\pi : \mathcal{X} \to \mathcal{U}$ by minimising a regression objective of the form:

$$\min_{\theta} \sum_{i=1}^{N} |\pi_\theta(\boldsymbol{x}_i) - \boldsymbol{u}_i|^2, \tag{2.27}$$

where $(\boldsymbol{x}_i, \boldsymbol{u}_i)_{i=1}^N$ are state–action pairs from expert demonstrations and θ are the policy parameters (Tedrake, 2023). In contrast, *implicit behaviour cloning* models the joint distribution of observations and actions, rather than directly fitting a deterministic policy. This formulation allows for sampling-based policy inference and includes approaches such as energy-based modeling (Florence et al., 2022), which has served as a precursor to recent diffusion-based policy learning methods. This problem often takes the form of *sequence learning*, where the objective is to model the temporal structure of expert behaviour (Chi et al., 2023; Zhao et al., 2023). Rather than predicting only the next action, these models learn to generate entire sequences of future actions conditioned on observed states, enabling control in an MPC-like manner. In both explicit and implicit variants, no explicit dynamics model or task specification is required; the policy is instead learned directly from demonstration data. Behaviour cloning has been successfully applied to a variety of manipulation tasks (Chi et al., 2023; Black et al., 2024; Zhao et al., 2023), particularly when high-quality demonstration data is available.

Reinforcement Learning More generally, learning can be extended beyond imitation to reinforcement learning approaches, where a policy is optimised through interaction with the environment rather than demonstrations. Reinforcement learning (RL) formulates control as a sequential decision-making problem in which

an agent learns a policy that maximizes expected cumulative reward through interaction with the environment (Sutton et al., 1998). This trial-and-error paradigm enables robots to acquire complex manipulation behaviors, such as pushing objects (Shetty et al., 2024) or performing in-hand manipulation (Andrychowicz et al., 2020; Handa et al., 2023). A key advantage of RL for contact-rich manipulation is its ability to explore stochastic rollouts without requiring explicit gradients of the reward function with respect to policy parameters. This is essential for learning behaviors that involve discontinuous contact dynamics, allowing to discover sequences of discrete contact modes by making and breaking contacts. Domain randomisation further improves robustness by exposing policies to randomised variations of physical parameters, such as friction, mass, and object geometry, during training (Andrychowicz et al., 2020; Muratore et al., 2022). By optimising over a distribution of environments (Haarnoja et al., 2018; Schulman et al., 2015, 2017), policies trained in simulation transfer better to real-world conditions and unseen objects. However, these benefits come at the cost of high sample complexity: RL typically requires many environment interactions to converge, making training computationally and data-intensive in practice. While this is often cited as a limitation of RL, it is worth noting that large behaviour cloning models can also demand substantial training resources. Moreover, despite showing some robustness, RL policies remain tightly coupled to the training domain and rarely generalize to new tasks or objects without retraining; unlike model-based planning and control approaches, whose adaptability is constrained primarily by the fidelity of their dynamics models rather than by prior experience.

Compounding Errors Both imitation and RL approaches can suffer from the problem of compounding errors, where small deviations from the expert behaviour can lead to states that are not well-covered by the training data, resulting in poor performance (Ross and Bagnell, 2010). Techniques like DAgger (Ross et al., 2011) mitigate this issue by iteratively collecting new data from the learned policy and querying the expert for corrective actions, thereby improving robustness and

generalisation. Yet, this only works under the assumption that (a) an expert is available for querying, and (b) that interactions with the real system are possible/safe during training, which is often not the case in practice. Alternative approaches have investigated to instead learn world-models from the expert data, which can then be used for model-based RL (DeMoss et al., 2023; Nematollahi et al., 2025). We note that the problem of compounding errors has also been extensively studied in the context of offline RL, for instance by Rigter et al. (2022). In the other direction, fixed behaviour-cloned policies can be further improved through techniques such as fine-tuning with online RL (Ankile et al., 2025; Yuan et al., 2024), though to date most successful demonstrations remain confined to simulation or low-dimensional tasks.

2.7.2 Model Learning

Rather than replacing model-based control approaches, learning can also be used to *improve* them. Model-based methods leverage explicit dynamics models, cost functions, and optimisation routines to plan and control robot behaviour. Learning can enhance each of these components: dynamics models can be learned from data when accurate physics models are unavailable (e.g., Hafner et al. (2019); Ai et al. (2025)); cost functions can be inferred from demonstrations (e.g., via inverse reinforcement learning (Finn et al., 2016)), and sampling-based optimisers can be accelerated by learning proposal distributions that guide the search toward promising regions of the action space (e.g. Sacks and Boots (2023); Power and Berenson (2024); Kurtz and Burdick (2025); Melon et al. (2020)). Such hybrid approaches preserve the structure and adaptability of model-based pipelines while improving efficiency and generalisation through data-driven components.

Relevance for This Thesis

While direct policy learning through imitation or reinforcement learning has shown remarkable progress in contact-rich manipulation, such approaches typically require large amounts of data and often struggle to generalise to new tasks or objects without retraining. In this thesis, we instead focus on a hybrid approach that integrates learning within the sampling-based

predictive control framework introduced in Section 2.5, thereby combining the adaptability of model-based control with the efficiency of learned components. By integrating learned generative models into optimisation-based control loops, we can enable controllers to implicitly reason about uncertainty while remaining amenable to extensions that treat uncertainty explicitly.

2.7.3 Generative Models for Control

Recent advances in generative modelling provide powerful tools for both direct policy learning and model-based control. In the following, we review three prominent classes of generative modelling that have been successfully applied to robot control: Variational Autoencoders (VAEs), Diffusion Models, and Flow-Matching.

Variational Autoencoders Variational Autoencoders (VAEs) (Kingma et al., 2019) are a class of generative models that encode input data $\boldsymbol{x}$ into a continuous latent space $\boldsymbol{z}$ and then decode from this space to reconstruct the original data. In the context of robot control, Conditional VAEs (CVAEs), which condition the generation process on additional context (e.g., current state or task parameters), have been successfully applied to learn structured representations of observation-action trajectories, such as in Action Chunking Transformer (ACT) (Zhao et al., 2023), or to learn sampling distributions over future actions (Ichter et al., 2018). A critical challenge arises when VAEs and CVAEs are applied to data with inherent multi-modality (i.e., data with distinct categories or modes, such as multiple possible future actions). This is primarily due to two architectural limitations. Firstly, standard VAEs, relying on simple Gaussian priors and a continuous latent space, tend toward mode-averaging. When the model attempts to synthesize data from an intermediate point between distinct modes, the optimisation of the reconstruction loss forces the decoder to generate a statistical average of these modes. In the context of image generation, this results in blurry image outputs that fail to capture the sharpness and distinctiveness of individual modes. Secondly, VAEs are susceptible to posterior collapse, a phenomenon where the encoder's posterior distribution

$q(\boldsymbol{z}|\boldsymbol{x})$ collapses to match the prior $p(\boldsymbol{z})$. To mitigate these issues, alternative generative architectures have been proposed. *Vector Quantized VAEs (VQ-VAEs)* address mode averaging by replacing the continuous latent space with a discrete codebook (Van Den Oord et al., 2017). This quantization step forces the model to select distinct, high-quality codes for each mode, thereby preventing continuous interpolation and yielding sharper, mode-specific samples.

Diffusion Models and Flow Matching These difficulties in modelling complex, multi-modal distributions highlighted the need for other generative frameworks capable of robustly capturing and sampling from disjoint data modes. This necessity has driven the adoption of novel approaches, notably *diffusion models* and *flow-matching*. These paradigms circumvent the challenges associated with variational bounds or adversarial training by transforming a simple prior distribution into the complex target data distribution through a series of small, reversible steps. This formulation inherently excels at resolving multi-modal ambiguities – as demonstrated by the success of models like *Diffusion Policy* (Chi et al., 2023), building on Diffusion Probabilistic Models (Ho et al., 2020); and π_0 (Black et al., 2024), building on flow-matching (Lipman et al., 2022) – in learning highly expressive, multi-modal conditional action distributions necessary for advanced visuomotor control tasks. While conceptually similar, flow-matching has recently shown to be more efficient at inference time than diffusion models for robot control tasks (Zhang and Gienger, 2024). The goal of flow-matching is to construct a flow that deterministically transports samples from a simple source distribution p (e.g., Gaussian noise) into a target distribution q (e.g., the data distribution). The framework is based on learning a velocity field $\boldsymbol{v}(t, x)$, which specifies how samples move through space over time. Each velocity field defines a time-dependent flow $\psi_t(x)$ by solving an ordinary differential equation (ODE), a process often referred to as simulation. In practice, flow-matching proceeds in two steps: first, a *probability path* $(p_t)_{0\leq t\leq 1}$ interpolating between p and q is designed, and second, a neural velocity field is trained to match the flow induced by this path. By integrating the learned ODE forward from $t = 0$

to $t = 1$, samples $X_0 \sim p$ are transformed into samples $X_1 = \psi_1(X_0)$ that follow the target distribution q. Through the lens of flow-matching, diffusion models can be interpreted as building a probability path via a *forward noising process* modelled by a particular type of stochastic differential equation (SDE) (Song et al., 2020). For detailed introductions to diffusion and flow-matching, and their theoretical connections we refer the reader to the tutorials by Nakkiran et al. (2024) and Lipman et al. (2024), respectively.

Relevance for This Thesis

In this thesis, we explore the use of generative models in Chapter 4, where they are employed as proposal distributions within a sampling-based MPC framework to improve the efficiency of online planning and control. We particularly focus on flow-matching, which offers efficient inference (superior to diffusion models), while being able to effectively capture the multi-modal action distributions characteristic of contact-rich manipulation tasks.

3

Handling Uncertainty through Reactive Sampling-Based Model Predictive Control

Publication Note

This chapter presents the work from the following publication (* indicates equal contribution):

Jankowski*, J., Brudermüller*, L., Hawes, N., and Calinon, S. (2023). VP-STO: Via-point-based stochastic trajectory optimization for reactive robot behavior. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10125–10131

Supplementary Material

Webpage with supplementary videos and code related to this chapter is available at: `https://sites.google.com/oxfordrobotics.institute/vp-sto`.

In this chapter, we introduce *Via-Point-based Stochastic Trajectory Optimisation* (VP-STO), a novel sampling-based model predictive control (MPC) algorithm that achieves high-frequency re-planning in high-dimensional, nonlinear systems. The summary in Table 1.2, repeated below, highlights that the primary objective of this work is to advance *reactivity* in robot control, handling uncertainty in dynamic environments *implicitly* through fast re-planning. This formulation assumes a *deterministic* and fully observable environment, where the underlying optimal control problem is cast as minimising costs in a receding-horizon framework. Sampling-based MPC has shown great promise for controlling complex robotic systems with nonlinear dynamics and non-convex cost functions (Williams et al., 2017; Bhardwaj

	Chapter 3	Chapter 4	Chapter 5	Chapter 6
Theme	Reactivity	Learning for Reactivity	Robustness	Exploration
Uncertainty Handling	Implicit	Implicit	Explicit	Explicit
Prior Knowledge of Uncertainty	None	None	Sampling-based access	Sampling-based access
Environment	Deterministic	Deterministic	Stochastic	Stochastic
Method	MPC	MPC	MPC	Belief-space control
Control Problem	Min. cost	Min. cost	Min. cost s.t. chance constraints	Max. information gain

Table 1.2: Theme, settings and methods for each chapter.

et al., 2022).Yet, its use in high-dimensional domains, such as robot manipulators, has been hindered by the computational burden of sampling and evaluating large numbers of candidate trajectories at high frequency. We argue that many of these limitations can be alleviated by adopting an efficient trajectory representation that supports fast sampling and optimisation in high-dimensional spaces. Crucially, such a representation enables planning directly in the *joint space* of a manipulator, typically far more complex than the *task space* (e.g., end-effector space) where most sampling-based MPC methods operate. Joint-space planning is essential for many manipulation tasks, as it exploits the robot's full kinematic capabilities, avoids singularities and joint limits, and allows the discovery of more diverse and physically feasible motions. This is particularly advantageous in contact-rich settings, where exploiting the full configuration space enables the controller to discover non-trivial and diverse motions that result in contact interactions with all parts of the robot, rather than just the end-effector. To this end, we propose a trajectory representation that is low-dimensional, time-optimal, and smooth by construction. Combined with gradient-free stochastic optimisation, this allows efficient optimisation over trajectory parameters, yielding adaptive, contact-rich manipulation behaviours in dynamic environments. While the true dynamics of the environment are unknown,

deterministic models suffice to generate motion plans that are continuously refined within a closed feedback loop. We validate this approach in both simulation and real-world experiments, demonstrating its effectiveness on non-prehensile pushing and dynamic grasping tasks with obstacles. Beyond the original work presented in Jankowski* et al. (2023), we have further extended VP-STO to include prior knowledge in the sampling process, enabling more efficient optimisation in contact-rich tasks (Jankowski et al., 2025a). We summarise these extensions at the end of this chapter in Section 3.1. Finally, we also provide ablation studies on the impact of the chosen number of via-points, as well as the Cholesky factorisation of the covariance matrix in CMA-ES, on the performance of VP-STO in Section 3.2. This was originally part of the supplementary material of Jankowski* et al. (2023), but we include it here for completeness.

VP-STO: Via-point-based Stochastic Trajectory Optimization for Reactive Robot Behavior

Julius Jankowski* [1,2], Lara Brudermüller* [3], Nick Hawes[3] and Sylvain Calinon[1,2]

***Abstract*— Achieving reactive robot behavior in complex dynamic environments is still challenging as it relies on being able to solve trajectory optimization problems quickly enough, such that we can replan the future motion at frequencies which are sufficiently high for the task at hand. We argue that current limitations in Model Predictive Control (MPC) for robot manipulators arise from inefficient, high-dimensional trajectory representations and the negligence of time-optimality in the trajectory optimization process. Therefore, we propose a motion optimization framework that optimizes *jointly* over space and time, generating smooth and timing-optimal robot trajectories in joint-space. While being task-agnostic, our formulation can incorporate additional task-specific requirements, such as collision avoidance, and yet maintain real-time control rates, demonstrated in simulation and real-world robot experiments on closed-loop manipulation.**

I. Introduction

In this paper we consider the problem of generating continuous, *timing-optimal* and smooth trajectories for robots operating in dynamic environments. Such task settings require the robot to be *reactive* to unforeseen changes in the environment, *e.g.*, due to dynamic obstacles, as well as to be *robust* and *compliant* when operating alongside or together with humans. However, generating this kind of reactive and yet efficient robot behavior within a high-dimensional configuration space is significantly challenging. This is especially the case in robot manipulation scenarios with many degrees of freedom (DoFs) as the resulting high-dimensional and multi-objective optimization problems are difficult to solve on-the-fly. A widespread approach in robotics is to formulate the task of motion generation as an optimization problem. Such *trajectory-optimization* based methods aim at finding a trajectory that minimizes a cost function, *e.g.*, motion smoothness, subject to constraints, *e.g.*, collision avoidance. Solution strategies can either be *gradient-based* or *sampling-based*. Approaches falling in the former category, *e.g.*, CHOMP [1] and TrajOpt [2], typically employ second-order iterative methods to find locally optimal solutions. However, they require the cost function to be once or even twice-differentiable, which constitutes a major limitation for manipulation tasks as they usually involve many complex, discontinuous cost terms and constraints. In contrast, sampling-based methods [3], [4] can operate on discontinuous costs by sampling candidate trajectories from a proposal distribution, evaluating them on the objective, and updating the proposal distribution according to their relative performance. Compared to gradient-based optimization, stochastic approaches typically also achieve higher robustness to difficult reward landscapes due to their exploratory properties [5]. Yet, achieving reactive robot behavior is challenging as it requires solving trajectory optimization problems at frequencies which are sufficiently high for the task at hand. This issue can be alleviated in *Model Predictive Control (MPC)* settings by optimizing over a shorter receding time-horizon. Stochastic, gradient-free trajectory optimization, such as Model-Predictive Path Integral (MPPI) control [6] and the Cross-Entropy-Method (CEM) [4], combined with MPC, also known as *sampling-based MPC*, has proven state-of-the-art real-time performance on real robotic systems in challenging and dynamic environments [7]–[9]. However, these works still suffer from limited long-term anticipation, *e.g.*, getting stuck in front of obstacles, due to the optimization over a short receding horizon.

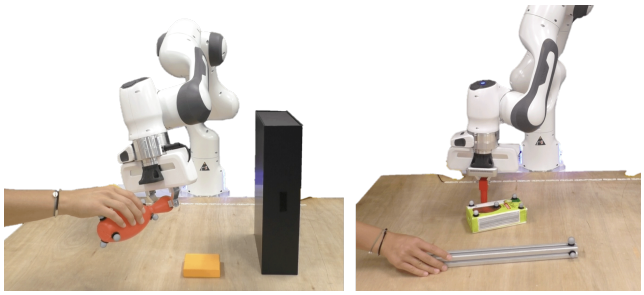


Fig. 1. ***Experiment settings.*** *Left:* Pick-and-place scenario, where the task is to grasp a bowling pin that is arbitrarily handed over to the robot and to place it upright in the middle of the table. *Right:* Pushing scenario, where the robot has to push the center of the green coffee packet to a moving target location indicated by the tip of the metal stick.

Motivated by the above, we propose *Via-Point-based Stochastic Trajectory Optimization (VP-STO)*, a framework that introduces the following contributions

1) A low-dimensional, time-continuous representation of trajectories in joint-space based on via-points that by-design respect kinodynamic constraints of the robot.
2) Stochastic via-point optimization, based on an evolutionary strategy, aiming at minimizing movement duration and task-related cost terms.
3) An MPC algorithm optimizing over the full horizon for real-time application in complex high-dimensional task settings, such as closed-loop object manipulation.

*Authors contributed equally.

JJ and SC were supported by the Swiss National Science Foundation (SNSF) through the CODIMAN project. LB was supported by an Amazon Web Services Lighthouse scholarship. NH received EPSRC funding via the "From Sensing to Collaboration" programme grant [EP/V000748/1].

[1]Idiap Research Institute, Martigny, CH; name.surname@idiap.ch
[2]Ecole Polytechnique Fédérale de Lausanne (EPFL), CH
[3]Oxford Robotics Institute, University of Oxford, UK; {larab, nickh}@robots.ox.ac.uk.

II. Related Work

In the context of closed-loop object manipulation with MPC, successful approaches to producing reactive robot behavior typically optimize in joint-space subject to kinodynamic constraints. While Fishman et al. use gradient-based MPC in order to find trajectories for human-robot handovers [10], a very recent approach named STORM [9] employed sampling-based MPC on robotic manipulation tasks. It is able to generate particularly smooth trajectories via low discrepancy action sampling, smooth interpolation and careful cost function design. Moreover, the parallelizability of sampling-based MPC is exploited by deploying the stochastic tensor optimization framework on a GPU. However, in contrast to our work, the approach relies on optimizing over a short receding horizon.

In the realm of time-parametrization of trajectories, most existing approaches fix the overall motion duration or do not specify it at all. For instance, the majority of MPC-based approaches only handle time implicitly via kinodynamic constraints. While the works of [11], [12] progress the state of the art in time-optimal MPC, their applicability to high-dimensional robotic systems yet is limited. In the context of motion planning, T-CHOMP [13] jointly optimizes a trajectory and the corresponding via-point timings. Yet, the total execution time is still fixed in advance. In contrast, Verscheure et al. consider time-optimal path tracking along a predetermined geometric path [14], where the timing along the path is optimized via a convex reformulation; however, the geometric path itself is fixed a priori. The way we approach the minimization of the movement duration is most similar to the work of [15]. However, in contrast to our work, their approach optimizes via-points and their timing separately.

III. Preliminaries: Trajectory Representation

The way we represent trajectories is based on previous work showing that the closed-form solution to the following optimization problem

$$\begin{aligned} &\min \quad \int_0^1 \boldsymbol{q}''(s)^\top \boldsymbol{q}''(s) ds \\ &\text{s.t.} \quad \boldsymbol{q}(s_n) = \boldsymbol{q}_n, \quad n = 1, ..., N \\ &\quad\;\; \boldsymbol{q}(0) = \boldsymbol{q}_0, \boldsymbol{q}'(0) = \boldsymbol{q}'_0,\; \boldsymbol{q}(1) = \boldsymbol{q}_T, \boldsymbol{q}'(1) = \boldsymbol{q}'_T \end{aligned} \tag{1}$$

is given by cubic splines [16] and that it can be formulated as a weighted superposition of basis functions [17]. Hence, the robot's configuration is defined as $\boldsymbol{q}(s) = \boldsymbol{\Phi}(s)\boldsymbol{w} \in \mathbb{R}^D$, with D being the number of degrees of freedom. The matrix $\boldsymbol{\Phi}(s)$ contains the basis functions which are weighted by the vector $\boldsymbol{w}$[1]. The trajectory is defined on the interval $\mathcal{S} = [0, 1]$, while the time t maps to the phase variable $s = \frac{t}{T} \in \mathcal{S}$ with T being the total duration of the trajectory. Assuming the trajectory duration T is given, the optimal trajectory can be written as

$$\boldsymbol{q}(t) = \boldsymbol{\Phi}\left(\frac{t}{T}\right)\boldsymbol{w}. \tag{2}$$

[1]A more detailed explanation of the basis functions and their derivation can be found in the appendix of [17].

The weight vector $\boldsymbol{w}$ for the basis functions includes the trajectory constraints consisting of the boundary condition parameters $\boldsymbol{w}_{\text{bc}} = [\boldsymbol{q}_0^\top, \boldsymbol{q}_0'^\top, \boldsymbol{q}_T^\top, \boldsymbol{q}_T'^\top]^\top$ and N via-points the trajectory has to pass through $\boldsymbol{q}_{\text{via}} = [\boldsymbol{q}_1^\top, \dots, \boldsymbol{q}_N^\top]^\top \in \mathbb{R}^{DN}$, such that $\boldsymbol{w} = [\boldsymbol{q}_{\text{via}}^\top, \boldsymbol{w}_{\text{bc}}^\top]^\top$. Accordingly, the optimal velocity and acceleration with respect to time are given by

$$\dot{\boldsymbol{q}}(t) = \frac{\partial \boldsymbol{q}(t)}{\partial t} = \frac{1}{T}\boldsymbol{\Phi}'\left(\frac{t}{T}\right)\boldsymbol{w}, \tag{3}$$

$$\ddot{\boldsymbol{q}}(t) = \frac{\partial^2 \boldsymbol{q}(t)}{\partial t^2} = \frac{1}{T^2}\boldsymbol{\Phi}''\left(\frac{t}{T}\right)\boldsymbol{w}, \tag{4}$$

where $f'(s)$ denotes differentiation w.r.t. the phase variable s, and $\dot{f}(t)$ differentiation w.r.t. time. Throughout this paper, the via-point timings s_n are assumed to be uniformly distributed in $\mathcal{S}$. Note that boundary velocities map to boundary derivatives w.r.t. s by multiplying them with the total duration T, *i.e.*, $\boldsymbol{q}'_0 = T\dot{\boldsymbol{q}}_0$ and $\boldsymbol{q}'_T = T\dot{\boldsymbol{q}}_T$. Furthermore, the optimization problem in Eq. (1) minimizes not only the objective $\boldsymbol{q}''(s)$, but also the integral over accelerations, since $\boldsymbol{q}''(s) = T^2\ddot{\boldsymbol{q}}(s)$ and thus the objective $\int_0^1 \ddot{\boldsymbol{q}}(s)^\top \ddot{\boldsymbol{q}}(s) ds$ directly maps to $\frac{1}{T^4}\int_0^1 \boldsymbol{q}''(s)^\top \boldsymbol{q}''(s) ds$, corresponding to the control effort. It is minimal *iff* the objective in Eq. (1) is minimal. As a result, this trajectory representation provides a linear mapping from via points, boundary conditions and the movement duration to a time-continuous and smooth trajectory.

In the remainder of the paper, we exploit this explicit parameterization with via-points and boundary conditions by optimizing only the via-points while keeping the predefined boundary condition parameters fixed. Thus, we write the computation of the trajectory as a superposition of a via-point term and a boundary constraints term, *i.e.*, $\boldsymbol{q}(s) = \boldsymbol{\Phi}_{\text{via}}(s)\boldsymbol{q}_{\text{via}} + \boldsymbol{\Phi}_{\text{bc}}(s)\boldsymbol{w}_{\text{bc}}$. The matrices $\boldsymbol{\Phi}_{\text{via}}(s)$ and $\boldsymbol{\Phi}_{\text{bc}}(s)$ are extracted from the basis function matrix $\boldsymbol{\Phi}(s)$.

IV. VP-Sto: Via-Point-Based Stochastic Trajectory Optimization

In the following, we introduce our stochastic trajectory optimization framework. The core idea is to find via-points $\boldsymbol{q}_{\text{via}}$ such that the synthesized trajectory minimizes a task-related objective, *i.e.*,

$$\min_{\boldsymbol{q}_{\text{via}}} \quad c\left[\boldsymbol{q}(s), \dot{\boldsymbol{q}}(s), \ddot{\boldsymbol{q}}(s), T\right]. \tag{5}$$

Based on these via-points, we efficiently synthesize high-quality trajectories, *i.e.*, $\boldsymbol{q}_{\text{via}} \rightarrow \boldsymbol{\xi}$ with $\boldsymbol{\xi} = \{\boldsymbol{q}(s), \dot{\boldsymbol{q}}(s), \ddot{\boldsymbol{q}}(s), T\}$. We aim at synthesizing trajectories that *by-design* minimize task-agnostic objectives, *i.e.*, *minimum time* and *smoothness*, and satisfy task-agnostic constraints, *i.e.*, equality constraints on the *initial* and *final state* and inequality constraints on *joint-space velocities* and *accelerations*. We employ stochastic black-box optimization, namely *Covariance Matrix Adaptation (CMA-ES)* [5] to optimize for the via-points. As each trajectory constructed from the sampled via-points already provides the optimal solution to the optimization problem given in Eq. 1, the CMA-ES optimization in the low-dimensional via-point space is

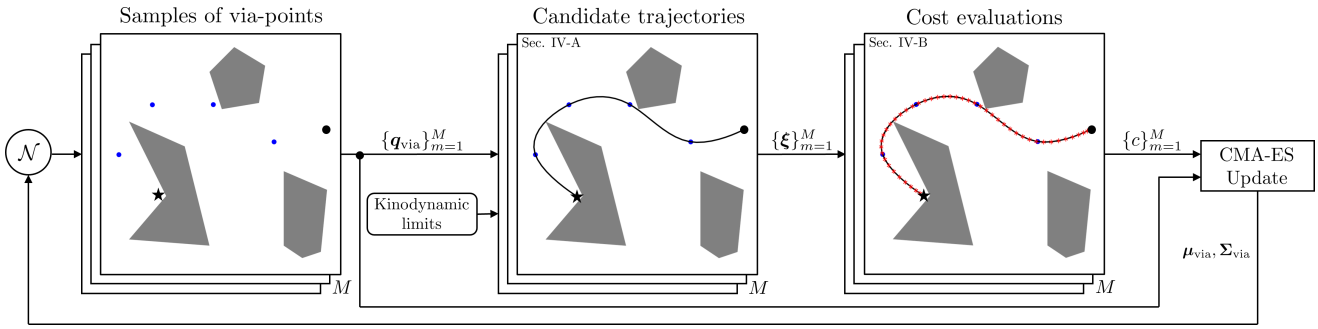


Fig. 2. An illustration of the via-point-based stochastic trajectory optimization loop. First, a new population of M via-points $\boldsymbol{q}_{\text{via}}$ is sampled from a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_{\text{via}}, \boldsymbol{\Sigma}_{\text{via}})$. Then, the sampled via-points are transformed into a population of candidate trajectories subject to kinodynamic limits. Next, the resulting trajectories are ranked according to their cost evaluations. Last, the parameters of the Gaussian sampling distribution are updated via CMA-ES using the cost rankings and the via-point sets themselves.

particularly fast, evaluating only high-quality trajectories. Moreover, with CMA-ES we are not only able to quickly converge to a local minimum, but to also leverage the exploration aspect of the evolutionary strategy (ES). In more detail, this nested optimization process, which is also illustrated in Fig. 2, comprises the following steps. First, a new population of M via-points $\boldsymbol{q}_{\text{via}}$ is sampled from a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_{\text{via}}, \boldsymbol{\Sigma}_{\text{via}})$. As $\boldsymbol{q}_{\text{via}}$ is a vector of the stacked via-points, note that $\boldsymbol{\mu}_{\text{via}} \in \mathbb{R}^{DN}$ and $\boldsymbol{\Sigma}_{\text{via}} \in \mathbb{R}^{DN \times DN}$. By taking M samples in this higher-dimensional space, instead of $M \cdot N$ samples for all via points separately in the configuration space, we are able to sample M sets of correlated via-points. Then, as described in detail in Sec. IV-A, the sampled via-points are transformed into a population of candidate trajectories that are evaluated according to cost terms as outlined in Sec. IV-B. Finally, we use CMA-ES in order to update the parameters $\boldsymbol{\mu}_{\text{via}}, \boldsymbol{\Sigma}_{\text{via}}$ of the Gaussian distribution of via-points. This optimization setup enables us to find a valid local minimum or even the global minimum at rates sufficient for reactive robot behavior in closed-loop manipulation tasks, as we demonstrate in our experiments outlined in Section VI.

A. Synthesis of Kinodynamically Admissible Trajectories

In this section, we show how sampled via-points $\boldsymbol{q}_{\text{via}}$ are translated into kinodynamically admissible trajectories. Up to this point, the trajectory has been represented in phase space as described in Sec. III. Given the via-points and the boundary conditions $[\boldsymbol{q}_0, \dot{\boldsymbol{q}}_0, \boldsymbol{q}_T, \dot{\boldsymbol{q}}_T]$, the explicit time-parameterized trajectory depends solely on the total movement duration T. This duration is determined by the dynamic limits on velocity $[\dot{\boldsymbol{q}}_{\min}, \dot{\boldsymbol{q}}_{\max}]$ and acceleration $[\ddot{\boldsymbol{q}}_{\min}, \ddot{\boldsymbol{q}}_{\max}]$, and is defined as the minimal positive duration for which the resulting velocity and acceleration profiles satisfy these limits:

$$\begin{aligned} T(\boldsymbol{q}_{\text{via}}) = \min_{\tau} \quad & \tau \\ \text{s.t.} \quad & \tau > 0, \\ & \dot{\boldsymbol{q}}_{\min} \leq \dot{\boldsymbol{q}}(s;\tau) \leq \dot{\boldsymbol{q}}_{\max}, \\ & \ddot{\boldsymbol{q}}_{\min} \leq \ddot{\boldsymbol{q}}(s;\tau) \leq \ddot{\boldsymbol{q}}_{\max}, \qquad \forall s \in [0,1]. \end{aligned} \tag{6}$$

To approximate T, we enforce the constraints only at a discrete set of phase values $\{s_k\}_{k=0}^{K}$, uniformly distributed in $\mathcal{S}$. For each evaluation point s_k, the velocity and acceleration constraints yield a closed-form minimal duration

$$T_k(\boldsymbol{q}_{\text{via}}),$$

computed directly from the profiles $\dot{\boldsymbol{q}}(s_k;\tau)$ and $\ddot{\boldsymbol{q}}(s_k;\tau)$. The minimal feasible duration over the entire trajectory is then conservatively approximated by

$$T(\boldsymbol{q}_{\text{via}}) = \max_k T_k(\boldsymbol{q}_{\text{via}}),$$

ensuring that all velocity and acceleration limits are satisfied at the chosen evaluation points. As a consequence of this construction, either the velocity or the acceleration profile reaches its limit at least at one evaluation point. Once T is determined, the explicit kinodynamically admissible trajectory $\boldsymbol{\xi}$ can be computed.

B. Cost Evaluation

Given the sampled and synthesized population of trajectories, we evaluate the performance, *i.e.*, the cost c of each trajectory, independently. The gradient-free optimizer allows for sharp cost function profiles, *e.g.*, trajectory constraints expressed through discontinuous barrier functions (cf. Sec. VI for examples). We approximate $c(\boldsymbol{\xi})$ by sampling the given trajectory with a predefined resolution Δs in the phase space $\mathcal{S}$ and accumulating the costs at these K evaluation points. In the time domain this can still map to varying resolutions of individual trajectories, as $\Delta t = T\Delta s$. Note that the evaluation points at s_k are not equivalent to the via-points at s_n, as depicted in Fig. 2. The resolution of s_k can be higher than that of s_n in order to have a better approximation of the trajectory cost while keeping the actual optimization variable $\boldsymbol{q}_{\text{via}}$ low-dimensional.

V. Online VP-STO (MPC)

In order to perform closed-loop control via continuous online re-optimization, we embed the *VP-STO* framework into an MPC algorithm. In this online setting, the main focus lies on rapidly finding valid movements connecting the current robot state $\boldsymbol{q}, \dot{\boldsymbol{q}}$ with a goal state $\boldsymbol{q}_T, \dot{\boldsymbol{q}}_T$ and re-optimizing

Algorithm 1: Online VP-STO: i-th MPC Step

Input: $\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{q}_T, \dot{\boldsymbol{q}}_T, \dot{\boldsymbol{q}}_{\min}, \dot{\boldsymbol{q}}_{\max}, \ddot{\boldsymbol{q}}_{\min}, \ddot{\boldsymbol{q}}_{\max}, \Delta t_{\text{mpc}}, T_{\text{stop}}, N_{\max}, \boldsymbol{\xi}^*_{i-1}$

Output: Short-horizon reference $\boldsymbol{q}_{\text{d}}(t), \dot{\boldsymbol{q}}_{\text{d}}(t), \ddot{\boldsymbol{q}}_{\text{d}}(t)$

$t_{\text{optimize}} \leftarrow 0$

$\boldsymbol{q}_0, \dot{\boldsymbol{q}}_0 \leftarrow \boldsymbol{q}, \dot{\boldsymbol{q}}$

$\boldsymbol{\xi}_{\text{direct}} \leftarrow$ `synthesize()` // V-A

if $\boldsymbol{\xi}_{direct}$ *is valid* **and** $\boldsymbol{\xi}_{direct}$ *is shorter than* T_{stop} **then**

  $\boldsymbol{\xi}^*_i \leftarrow \boldsymbol{\xi}_{\text{direct}}$

else

  if $\boldsymbol{\xi}^*_{i-1}$ *is valid* **then**

    ${}^0\boldsymbol{\mu}_{\text{via}}, {}^0\boldsymbol{\Sigma}_{\text{via}}, N \leftarrow$ `warmStart`$(\boldsymbol{\xi}^*_{i-1})$ // V-B

  else

    ${}^0\boldsymbol{\mu}_{\text{via}}, {}^0\boldsymbol{\Sigma}_{\text{via}}, N \leftarrow$ `exploreInit()` // V-B

  end

  $j \leftarrow 0$

  while $t_{optimize} < \Delta t_{mpc}$ **do**

    $\{\boldsymbol{q}_{\text{via}}\}_{m=1}^{M} \leftarrow$ `sample`$\left({}^j\boldsymbol{\mu}_{\text{via}}, {}^j\boldsymbol{\Sigma}_{\text{via}}\right)$ // V-C

    $\{\boldsymbol{\xi}\}_{m=1}^{M} \leftarrow$ `synthesize`$\left(\{\boldsymbol{q}_{\text{via}}\}_{m=1}^{M}\right)$ // IV-A

    $\{c\}_{m=1}^{M} \leftarrow$ `evaluate`$\left(\{\boldsymbol{\xi}\}_{m=1}^{M}\right)$ // IV-B

    $\boldsymbol{\mu}_{\text{via}}^{j+1}, \boldsymbol{\Sigma}_{\text{via}}^{j+1} \leftarrow$ `sep-CMA-ES`$\left(\{\boldsymbol{q}_{\text{via}}, c\}_{m=1}^{M}\right)$

    $j \leftarrow j + 1$

  end

  $\boldsymbol{\xi}^*_i \leftarrow$ `synthesize`$\left(\boldsymbol{\mu}_{\text{via}}^{j}\right)$

end

$\boldsymbol{q}_{\text{d}}(t), \dot{\boldsymbol{q}}_{\text{d}}(t), \ddot{\boldsymbol{q}}_{\text{d}}(t) \leftarrow$ `shortHorizon`$(\boldsymbol{\xi}^*_i)$ // V-D

them at a sufficient rate $f_{\text{mpc}} = \frac{1}{\Delta t_{\text{mpc}}}$. Algorithm 1 outlines a single MPC step that, given the current robot state, attempts to find an optimal *full-horizon* trajectory and to extract a short-horizon reference to be tracked by a lower-level impedance controller. The details of the algorithm will be outlined in the remainder of this section.

In the online setting the number of via-points N used to parameterize the trajectory plays an important role. A large number of via-points can capture highly complex movements and may find more optimal solutions. However, it also implies a higher-dimensional decision space which increases the computational complexity of the optimization loop. Consequently, a particular focus within the MPC algorithm lies on the selection of N.

A. No-Via-Point Trajectory for Stopping Behavior

VP-STO is based on optimizing the locations of a given number of via-points. However, the trajectory synthesis, described in Sec. IV-A, also works without any via-points, *i.e.*, N=0. The resulting trajectory connects the current robot state and the desired state by a third-order polynomial that minimizes the smoothness objective in Eq. (1) and satisfies the kinodynamic limits. As this no-via-point trajectory is a unique solution, it can not account for any other movement objectives, *e.g.*, to avoid collisions. Yet, the advantage is a cheap-to-construct trajectory that has no stochasticity, which is useful for driving the robot to the target configuration and stopping with zero velocity. Therefore, at the beginning

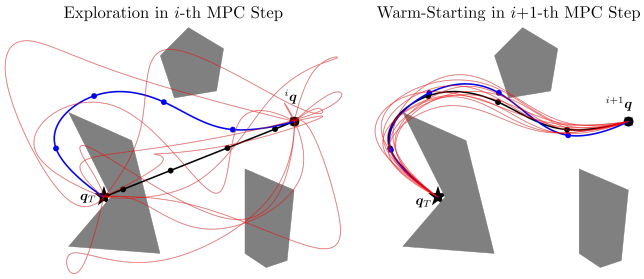


Fig. 3. An illustration of the stochastic optimization process within the proposed MPC algorithm. ***Left:*** In the *exploration* mode, trajectories are sampled and synthesized with a large initial variance in order to discover valid solutions. ***Right:*** If a valid solution is available from the previous MPC step, we *warm-start* the optimization by shifting the solution and sampling from a lower-variance initial distribution. All sampled trajectories are shown in red. The initial guesses ${}^0\boldsymbol{\mu}_{\text{via}}$ of an MPC step are depicted by the black solid lines, while the blue trajectories illustrate the mean solution ${}^{20}\boldsymbol{\mu}_{\text{via}}$ after 20 optimization iterations.

of each optimization cycle, we first check if this simple direct trajectory is valid, *e.g.*, collision-free, and if the corresponding duration of the movement is below the user-defined threshold T_{stop}. By setting the threshold rather small, we let the mechanism take over towards the final part of the total trajectory to achieve robust stopping behavior for reaching the goal. If the direct solution is not used, we perform a *VP-STO* optimization cycle.

B. Initialization: Exploration vs. Warm-Starting

The use of an evolutionary optimization strategy, such as CMA-ES, allows us to initialize the optimization not only with an initial guess of the via-points $\boldsymbol{\mu}_{\text{via}}$, but also to set the corresponding initial variance $\boldsymbol{\Sigma}_{\text{via}}$ as an estimate of how certain we are about the initial solution. The initial variance can thus be interpreted as an exploration parameter influencing how the very first population of candidate trajectories will be sampled. Therefore, in each MPC step we use two possible modes on how to initialize these parameters. The effects of each mode on the resulting candidate trajectories are shown in Fig. 3.

Exploration. If a MPC step was not successful in finding a valid trajectory, the successive MPC step will be used to *explore* a larger area of the trajectory space to ideally discover a valid solution, as can be seen from the sampled trajectories in the left of Fig. 3. We initialize the mean solution $\boldsymbol{\mu}_{\text{via}}$ with a naive straight-line guess with high uncertainty, *i.e.*, large diagonal values of $\boldsymbol{\Sigma}_{\text{via}}$. The number of via-points used to parameterize the trajectory is set to $N = N_{\max}$, with $N_{\max}$ being specified by the user and depends on the complexity of the task, as well as on the available computational resources.

Warm-Starting. If a valid solution was found in a MPC step, we shift the solution forward in time and use it to *warm-start* the mean $\boldsymbol{\mu}_{\text{via}}$ in the successive MPC step, potentially further improving the current solution. In this case, we initialize the covariance matrix $\boldsymbol{\Sigma}_{\text{via}}$ with low values on the diagonal as we are more certain about the proximity of the current

solution to a valid local minimum, as can be seen on the right of Fig. 3. In order to determine the number of via-points N for the successive MPC step, we use the movement duration of the current solution as a proxy for how complex the remainder of the movement will be. We therefore set $N = \max(1, \min(\lceil \alpha T \rceil, N_{\max}))$, where T is the total duration of the current solution and α a user-defined scaling parameter.

C. Efficient Gaussian Sampling of Smooth Trajectories through Covariance Matrix Decomposition

For the sake of computational efficiency and linear scalability to high-DoF systems in our MPC solver, we use a variant of CMA-ES that iterates on diagonal covariance matrices instead of full covariance matrices, namely *sep-CMA-ES* [18]. However, a diagonal covariance matrix does not capture the correlations between the sampled via-points that are important for sampling smooth trajectories. We counteract this disadvantage by using a Cholesky factorization of the covariance matrix, such that $\boldsymbol{\Sigma}_{\text{via}} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^\top$, where the diagonal matrix $\boldsymbol{D} = \text{diag}(\boldsymbol{\sigma}_{\text{via}})$ is subject to iterative optimization through sep-CMA-ES. This renders our algorithm to a computational complexity of $\mathcal{O}(ND)$, with N being the number of via-points and D the DoF of the robot, instead of $\mathcal{O}((ND)^2)$ in the case of the full covariance matrix. The lower triangular matrix $\boldsymbol{L}$ is computed offline as the Cholesky decomposition of a constant covariance matrix

$$\boldsymbol{\Sigma}_{\text{smooth}} = \boldsymbol{L}\boldsymbol{L}^\top = \left(\int_0^1 \boldsymbol{\Phi}''_{\text{via}}(s)^\top \boldsymbol{\Phi}''_{\text{via}}(s) ds \right)^{-1}, \quad (7)$$

that is derived from a probability distribution of smooth trajectories, *i.e.*, $p_{\text{smooth}}(\boldsymbol{q}_{\text{via}}, \boldsymbol{w}_{\text{bc}}) \propto \exp(-c_{\text{effort}})$, with $c_{\text{effort}} = \frac{1}{2} \int_0^1 \boldsymbol{q}''(s)^\top \boldsymbol{q}''(s) ds$.

D. Impedance Control

At a lower control level, we deploy an impedance controller that runs at a control rate of 1 kHz, which requires a finely sampled reference trajectory. Due to our time-continuous representation of the optimized trajectory, we can sample configurations from it with arbitrarily small temporal resolution. Each MPC step yields an optimized trajectory $\boldsymbol{\xi}_i^*$, from which we extract a position-, velocity- and acceleration-reference enabling the robot to track the current movement plan.

VI. Experiments

We evaluate the effectiveness and performance of the *VP-STO* framework in simulation, as well as in real-world experiments with a Franka Emika robot arm.

A. Simulation

We begin by evaluating our framework in an offline planning setting for a 2D point mass in a cluttered toy environment adopted from [9]. In this experiment, we run *VP-STO* (cf. Sec. IV) for 100 times with a straight-line initialization. The left plot in Fig. 4 shows the resulting 100 trajectories after convergence. The majority of the found solutions converged to 3 valid local optima, *i.e.*, 28 solutions

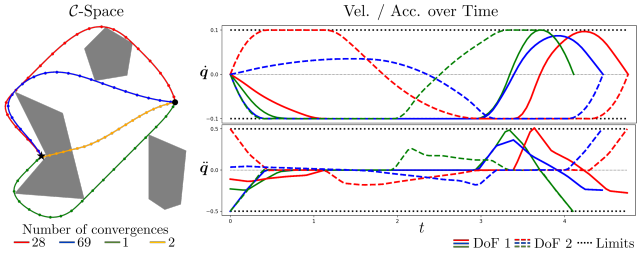


Fig. 4. ***Offline VP-STO.*** *Left:* The resulting trajectories from 100 experiment runs when initializing with a straight-line guess between the start position (black circle) and the target position (black asterisk). The number of convergence indicates how often *VP-STO* converged to the corresponding color-coded solution. *Right:* The velocity and acceleration profiles for each degree of freedom corresponding to the valid solutions on the left.

to the red, 69 to the blue and one to the green trajectory. Only 2 runs produce a non-valid solution, shown in yellow. We note here that gradient-based trajectory optimization methods given the straight-line initial guess in such a challenging environment would only converge to this non-valid local optimum. Moreover, this also shows that the choice of CMA-ES as a solver for our framework helps to converge to the present local optima with negligible error, despite the stochasticity in the sampling of the via-points. Last, the corresponding velocity and acceleration profiles (only shown for the valid solutions), depicted on the right of Fig. 4, reflect the timing-optimal property of the generated trajectories. After applying maximum acceleration at the start of the movement, the robot moves at maximum speed within the limits before it again applies the maximum acceleration to stop at the goal. This implies that our framework generates trajectories that not only respect the given dynamic limits, but also exploits them in the spirit of timing-optimality.

For the online setting, as described in Sec. V, we compare *VP-STO* to *STORM* [9], which we consider as state-of-the-art in sampling-based MPC for producing reactive robot behavior. Again using the scenario from above, we run 5 experiments in which we deploy *VP-STO* within the MPC-algorithm (cf. Alg. 1). The resulting trajectories are shown in blue in Fig. 5 alongside the 5 solutions in red generated by *STORM*. It can be seen that *STORM* is not able to reach the goal. Especially, due to the short-horizon optimization scheme, the robot first follows the path with the shortest distance towards the goal while not being able to anticipate moving around the obstacle early enough. Therefore, it gets stuck in front of the obstacle. In contrast, *Online VP-STO* produces solutions which allow the robot to smoothly navigate to the goal, while exploiting its velocity and acceleration limits. The given setting and experiment emphasizes the advantage of our efficient formulation which allows us to always optimize over the full horizon.

B. Real-World Experiments

We demonstrate *VP-STO* on a real robot using the manipulation scenarios in Fig. 1: a *pick-and-place* and a *box pushing* task. We increase the complexity of both scenarios by disturbing the robot and the target objects. This requires

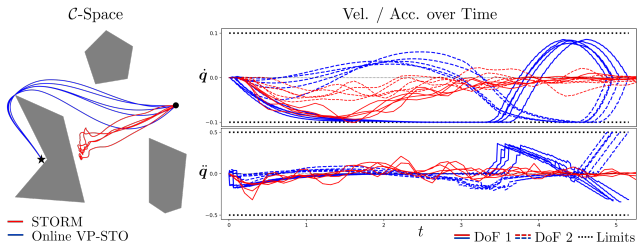


Fig. 5. ***Online VP-STO (MPC).*** *Left:* The trajectories taken by the robot when deploying *VP-STO* in an MPC setting (blue), as opposed to using *STORM* [9] (red). *Right:* The velocity and acceleration profiles for each degree of freedom corresponding to the found solutions on the left.

a fast feedback loop provided by *Online VP-STO*.

Setup. Both experiments are performed on a Franka Emika robot arm. The framework was run on Ubuntu 20.04 with an Intel Core i7-8700 CPU@3.2GHz and 16GB of RAM. The poses of the objects were tracked with a Vicon motion capture system and post-processed with an extended Kalman filter. The MPC steps are executed at a fixed control rate (specified below). In a single MPC step, we run optimization iterations until the next MPC step starts.

a) Pick-and-Place: First, we consider a *pick-and-place* scenario under human intervention. The robot's task is to grasp a pin, *i.e.*, the picking phase, and to place it in an upright position in a given target location in the workspace, *i.e.*, the placing phase. In the picking phase, the pin can be either handed over to the robot in arbitrary poses or the robot needs to pick it up from the table. This phase requires real-time collision avoidance in narrow configuration passages, *i.e.*, the robot has to avoid collisions between its hand, including the fingers, and the pin while reaching a configuration where the hand encloses the pin. For the grasp pose, we run a separate pose optimization process in parallel to *VP-STO*, providing the final robot configuration $\boldsymbol{q}_T$. After a successful grasp, the robot continues with the placing phase. The challenge here is that the pin might still move within the gripper due to its own weight or due to interference from a user. Consequently, feedback of the current pin pose is needed to avoid collisions between the pin and the environment and to correctly place the pin. We parameterize the sampled trajectories with a maximum number of via-points N_{max}=4 and α=2. *VP-STO* replans with a rate of 12.5 Hz.

b) Box Pushing: In the second scenario, we address the task of planning and control through physical contacts, *i.e.*, the robot is supposed to push a box towards a moving target position. Such a task requires the robot to deliberately make and break contacts, which is subject to discontinuous cost-landscapes. Here, we exploit the presented trajectory parameterization by setting the final robot configuration $\boldsymbol{q}_T$ of each MPC step such that the end-effector moves towards the center of the box. This enforces all sampled candidate trajectories to make contact with the box. The point of contact and the resulting dynamics of the box depends on the location of the via-points which are subject to minimizing the distance between the box position and the target.For the sake of fast simulations of the contact dynamics, we use a quasi-dynamic model for the box dynamics parallel to the table surface. *VP-STO* is executed with a constant number of via-points N=3 at a control rate of 20 Hz.

Cost Terms. We begin with the task-agnostic terms and conclude with more task-specific terms.

Movement Duration: The movement duration is used explicitly as part of the cost function in order to minimize the time needed for the remaining robot movement.

Smoothness: In order to optimize not only for fast, but also efficient movements, we use the same metric as in Eq. (1) as the smoothness cost term.

Joint Limit Avoidance: For keeping the robot configuration inside the joint angle limits, we deploy a discontinuous metric that accounts for joint limit violations, *i.e.*,

$$c_{\text{jla}}(q) = \begin{cases} 1 + q - q_{\max}, & \text{if } q \geq q_{\max} \\ 1 + q_{\min} - q, & \text{if } q \leq q_{\min} \\ 0, & \text{otherwise} \end{cases}. \tag{8}$$

We consider a trajectory to be invalid if it results in a joint limit violation, *i.e.*, $q \geq q_{\max}$ or $q \leq q_{\min}$.

Collision Avoidance: In order to efficiently evaluate the validity of a trajectory regarding collisions between the robot and the environment, we perform binary collision checks for each configuration evaluated along the trajectory, instead of computing a distance between two geometries. Thus, the collision cost for a single trajectory is equal to the number of evaluation points that are in collision. Similarly to the joint limit avoidance cost, we consider a trajectory to be invalid if it results in a collision.

Pushing Progress: In the case of a pushing task, we further require a cost term that rewards trajectories which let the robot move the box closer to the current desired target $\boldsymbol{x}_{\text{box}}^{\text{des}}$. We evaluate the pushing progress of a single trajectory by first simulating the contact dynamics that result in a trajectory of the box $\boldsymbol{x}_{\text{box}}(t)$; and then computing the box position error at the beginning $e_{\text{box},0} = ||\boldsymbol{x}_{\text{box}}(0) - \boldsymbol{x}_{\text{box}}^{\text{des}}||_2^2$ and at the end $e_{\text{box},T} = ||\boldsymbol{x}_{\text{box}}(T) - \boldsymbol{x}_{\text{box}}^{\text{des}}||_2^2$ of the robot movement. The final pushing progress cost is given by $c_{\text{push}}(\boldsymbol{\xi}) = \exp(e_{\text{box},T} - e_{\text{box},0})$. Additionally, we consider trajectories that move the box away from the target, *i.e.*, $e_{\text{box},T} \geq e_{\text{box},0}$, to be invalid. In that case, the *exploration* mode in the next MPC step (cf. Sec. V-B) is triggered.

Results. First, we note that throughout the experiments, the robot did not collide with any objects in the workspace and did not violate the joint limits. When the experimenter perturbs the robot, *i.e.*, disturbing it through physical interaction or pulling the pin out of the gripper, the robot is compliant and adapts its motion. In the pick-and-place scenario, it robustly picked up the pin from various locations in the workspace, including handovers by the experimenter; and placed it at the desired target location in all runs. In the box pushing scenario, the robot manages to find pushing motions from arbitrary configurations and box locations and to eventually push the box into the target. We note, however, that some changes of the target location resulted in the robot not finding a valid pushing motion quickly

enough, which in turn made the robot push the box out of the workspace. This could only be recovered by the experimenter. Recordings of the experiments and additional material can be found in the accompanying video to this paper and on the dedicated website https://sites.google.com/oxfordrobotics.institute/vp-sto.

VII. Conclusion

We presented a motion optimization framework that is able to generate reactive and yet smooth and efficient robot behavior for complex high-dimensional robot tasks. In contrast to standard trajectory optimization techniques, sampling-based and gradient-based, our framework outputs trajectories which not only optimize over space but also time. Moreover, due to the full-horizon optimization in an MPC-setting, it is particularly suitable for closed-loop manipulation tasks that demand for continuous re-planning and feedback. We successfully demonstrate this in two real-world experiments on a Franka Emika robot arm, *i.e.*, a pick-and-place and a box-pushing scenario.

We wish to extend and improve our work by considering the following points. First, the number via-points to sample yet is subject to heuristic tuning. In general, with increasing movement complexity more via-points are needed at the cost of higher computational complexity. Future work should make the selection of this hyper-parameter more intuitive. And second, we would like to further increase the robustness of *VP-STO* by considering uncertainties in the interaction between the robot and its environment. This includes to explore stochastic roll-outs in the cost evaluation.

References

[1] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "CHOMP: Covariant Hamiltonian optimization for motion planning," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.

[2] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[3] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 4569–4574.

[4] R. Y. Rubinstein, "The Cross-Entropy Method for Combinatorial and Continuous Optimization," *Methodology and Computing in Applied Probability*, vol. 1, no. 2, pp. 127–190, 1999.

[5] N. Hansen, "The CMA evolution strategy: A tutorial," *arXiv preprint arXiv:1604.00772*, 2016.

[6] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control: From theory to parallel computation," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 344–357, 2017.

[7] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1433–1440.

[8] M. Bangura and R. Mahony, "Real-time Model Predictive Control for Quadrotors," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11 773–11 780, 2014.

[9] M. Bhardwaj, B. Sundaralingam, A. Mousavian, N. Ratliff, D. Fox, F. Ramos, and B. Boots, "STORM: An Integrated Framework for Fast Joint-Space Model-Predictive Control for Reactive Manipulation," in *Conference on Robot Learning (CoRL)*, 2022, pp. 750–759.

[10] A. Fishman, C. Paxton, W. Yang, D. Fox, B. Boots, and N. Ratliff, "Collaborative interaction models for optimized human-robot teamwork," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 11 221–11 228.

[11] L. Van den Broeck, M. Diehl, and J. Swevers, "Model predictive control for time-optimal point-to-point motion control," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 2458–2463, 2011.

[12] C. Rosmann, A. Makarow, F. Hoffmann, and T. Bertram, "Time-Optimal nonlinear model predictive control with minimal control interventions," in *Proc. IEEE Conference on Control Technology and Applications (CCTA)*, 2017, pp. 19–24.

[13] A. Byravan, B. Boots, S. S. Srinivasa, and D. Fox, "Space-time functional gradient optimization for motion planning," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 6499–6506.

[14] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2318–2327, 2009.

[15] M. Toussaint, J. Harris, J.-S. Ha, D. Driess, and W. Hönig, "Sequence-of-Constraints MPC: Reactive Timing-Optimal Control of Sequential Manipulation," *arXiv preprint arXiv:2203.05390*, 2022.

[16] Z. Zhang, J. Tomlinson, and C. Martin, "Splines and linear control theory," *Acta Math. Appl*, vol. 49, pp. 1–34, 1997.

[17] J. Jankowski, M. Racca, and S. Calinon, "From Key Positions to Optimal Basis Functions for Probabilistic Adaptive Control," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3242–3249, 2022.

[18] R. Ros and N. Hansen, "A simple modification in cma-es achieving linear time and space complexity," in *International conference on parallel problem solving from nature*. Springer, 2008, pp. 296–305.

APPENDIX

A. Efficient Gaussian Sampling of Smooth Trajectories through Covariance Matrix Decomposition (Extended)

In the proposed MPC algorithm, the number of optimization iterations ran in a single step is limited by the desired control rate and the computational resources. We have identified two modifications of the algorithm that drastically reduce the cost of the mean trajectory after a given optimization time budget.

First, we replace the standard CMA-ES optimization by sep-CMA-ES, a variant that iterates only on diagonal covariance matrices. This improves the computational complexity from $\mathbb{O}(N^2D^2)$ (CMA-ES) to $\mathbb{O}(ND)$ (sep-CMA-ES), meaning that the computational load of sampling from and updating the covariance matrix scale linearly with the number of via-points N and the DoF D of the robot.

Second, instead of initializing the covariance matrix $\boldsymbol{\Sigma}_{\text{via}}$ with an identity matrix scaled by a single scalar, we start the optimization with a covariance matrix that captures smoothness correlations between via-points. This modification can be justified by a probabilistic view on stochastic optimization problems, *i.e.*, rather than minimizing the expected cost $c(\boldsymbol{q}_{\text{via}})$ as in (5), we aim at maximizing a probability $p(\boldsymbol{q}_{\text{via}}) \propto e^{-c(\boldsymbol{q}_{\text{via}})}$. It is easy to show that both optimization problems have equivalent optima. In fact, CMA-ES attempts to locally approximate the generally intractable probability distribution $p(\boldsymbol{q}_{\text{via}})$ by a Gaussian distribution in each iteration. If the trajectory cost is given as a sum of multiple objectives, *i.e.*, $c(\boldsymbol{q}_{\text{via}}) = \sum_i c_i(\boldsymbol{q}_{\text{via}})$, the corresponding probability distribution can be written as a product of multiple probability distributions, *i.e.*, $p(\boldsymbol{q}_{\text{via}}) \propto \prod_i e^{-c_i(\boldsymbol{q}_{\text{via}})}$. A smoothness metric is typically part of the cost function, in our case we use

$$\begin{aligned} c_{\text{smooth}}(\boldsymbol{q}_{\text{via}}) &= \frac{1}{2}\int_0^1 \boldsymbol{q}''(s)^\top \boldsymbol{q}''(s) ds \\ &= \frac{1}{2}\boldsymbol{w}^\top \int_0^1 \boldsymbol{\Phi}''(s)^\top \boldsymbol{\Phi}''(s) ds \boldsymbol{w}. \end{aligned} \tag{9}$$

Since $\boldsymbol{w} = [\boldsymbol{q}_{\text{via}}^\top, \boldsymbol{w}_{\text{bc}}^\top]^\top$, the smoothness cost term can be exactly represented by a Gaussian distribution, *i.e.*,

$$p_{\text{smooth}}(\boldsymbol{q}_{\text{via}}, \boldsymbol{w}_{\text{bc}}) = \mathcal{N}\left(\boldsymbol{0}, \int_0^1 \boldsymbol{\Phi}''(s)^\top \boldsymbol{\Phi}''(s) ds\right). \tag{10}$$

We condition the joint distribution on the given boundary constraints to obtain the corresponding distribution of the via-points $p_{\text{smooth}}(\boldsymbol{q}_{\text{via}}|\boldsymbol{w}_{\text{bc}}) = \mathcal{N}(\boldsymbol{\mu}_{\text{via, smooth}}, \boldsymbol{\Sigma}_{\text{via, smooth}})$ with

$$\begin{aligned} \boldsymbol{\mu}_{\text{via, smooth}} &= \boldsymbol{\Sigma}_{\text{via, smooth}} \int_0^1 \boldsymbol{\Phi}''_{\text{via}}(s)^\top \boldsymbol{\Phi}''_{\text{bc}}(s) ds \; \boldsymbol{w}_{\text{bc}} \\ \boldsymbol{\Sigma}_{\text{via, smooth}} &= \left(\int_0^1 \boldsymbol{\Phi}''_{\text{via}}(s)^\top \boldsymbol{\Phi}''_{\text{via}}(s) ds\right)^{-1}. \end{aligned} \tag{11}$$

By initializing the covariance matrix with $\boldsymbol{\Sigma}_{\text{via, smooth}}$, the very first population of via-points that is evaluated in an optimization loop is consequently sampled from p_{smooth}. This can be interpreted as an informed warm-starting of the covariance matrix in a CMA-ES loop.

In order to integrate the off-diagonal structure of $\boldsymbol{\Sigma}_{\text{via, smooth}}$ with the diagonal covariance matrix $\text{diag}(\boldsymbol{\sigma}_{\text{via}})$ that is updated by sep-CMA-ES, we assemble the final covariance matrix by a Cholesky factorization, *i.e.*, $\boldsymbol{\Sigma}_{\text{via}} = \boldsymbol{L}\text{diag}(\boldsymbol{\sigma}_{\text{via}})\boldsymbol{L}^\top$. The off-diagonal structure is imposed by the lower triangular matrix $\boldsymbol{L}$ that is given by the Cholesky decomposition of the smoothness covariance, such that $\boldsymbol{\Sigma}_{\text{via, smooth}} = \boldsymbol{L}\boldsymbol{L}^\top$.

B. Ablation Studies

In the paper, we present design choices that we want to further justify via ablation studies.

1) Impact of the Number of Via-points: In this ablation study, we investigate the impact of the number of via-points used to represent the robot movement. This hyper-parameter has a high impact on the overall framework performance. On one hand, it directly sets the dimensionality of the optimization problem to solve; on the other hand, it directly spans the space of movements that can be synthesized. From an optimization perspective, tuning the number of via-points gives us an intuitive way of increasing/decreasing resources on an optimization result with a decreasing/increasing cost. We illustrate this relationship in Fig. 6, where we let a 1D double-integrator move from $q_0 = 0.0$, $\dot{q}_0 = 0.0$ to $q_T = 1.0$, $\dot{q}_T = 0.0$ in minimal time, subject to a maximum velocity $|\dot{q}| < 0.1$ and an acceleration limit $|\ddot{q}| < 0.2$; with a varying number of via-points. This time-optimal control problem is known to be solved by a bang-bang acceleration profile, such that we know the analytic limit of the minimal time to be $c_{\text{bang-bang}} = T_{\text{bang-bang}} = 10.5$, which is depicted as dashed black line in the upper-left plot. We observe that the solution cost exponentially converges to $c_{\text{bang-bang}}$ as we increase the number of via-points. The lower-left plot shows the number of CMA-ES-iterations required to converge as a function of the number of via-points. Here, we detect convergence if $|c_k - c_{k-1}| < 10^{-6}$ in the k-th iteration. Interestingly, the number of iterations grows linearly with the number of via-points. Note that this does not mean that the computational cost grows linearly with the number of via-points, since the computational cost for a single iteration is either linear (sep-CMA-ES) or quadratic (CMA-ES) in the number of via-points. Nevertheless, those results motivate to use a low number of via-points as with a growing number of via-points, the benefit of adding a via-point is not worth the extra computational cost.

2) Impact of the Cholesky Factorization of the Covariance Matrix: In this ablation study, we look at a 2D minimal-time planning problem including an obstacle that is to be avoided. We fix the number of via-points to $N = 6$ and set up four different optimization loops that are supposed to solve the same problem. Each setup uses either CMA-ES or sep-CMA-ES and runs with or without the Cholesky factorization of the covariance matrix as described in Sec. V-C. For comparison, we look at the cost evolution over the number of iterations. The dashed black line in all plots

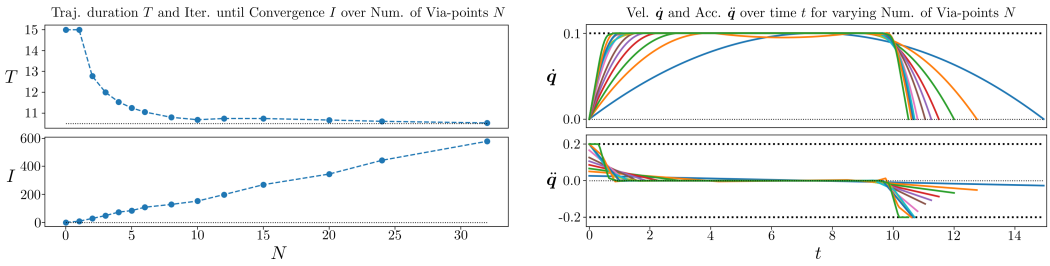


Fig. 6. A study of the impact of the number of via-points in a 1D time-optimization problem. **Top-Left:** Impact on the resulting movement duration. The dotted black line illustrates the duration of the optimal *bang-bang* solution. **Bottom-Left:** Impact on the number of iterations required until convergence. **Right:** Velocity and acceleration profiles for evaluated numbers of via-points.

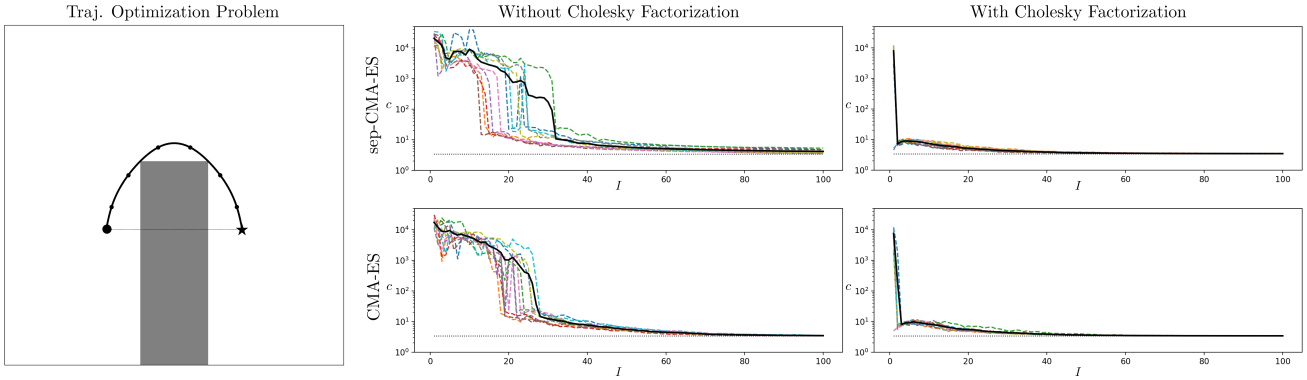


Fig. 7. A study of the impact of the Cholesky factorization of the Covariance Matrix $\boldsymbol{\Sigma}_{\text{via}}$ in a 2D time-optimization problem with obstacle avoidance. **Left:** The configuration space including the obstacle in gray, the initial guess as dashed line, and the optimal solution around the obstacle as solid line together with the corresponding via-points as circles. **Center:** The via-point covariance matrix is explicitly updated, *i.e.*, $\boldsymbol{\Sigma}_{\text{via}} = \boldsymbol{\Sigma}_{\text{CMA}}$. **Right:** The via-point covariance matrix is updated through a Cholesky factorization, *i.e.*, $\boldsymbol{\Sigma}_{\text{via}} = \boldsymbol{L}\boldsymbol{\Sigma}_{\text{CMA}}\boldsymbol{L}^{\top}$. **Top:** sep-CMA-ES iterates on diagonal covariance matrices only, *i.e.*, $\boldsymbol{\Sigma}_{\text{CMA}} = \text{diag}(\boldsymbol{\sigma}_{\text{CMA}})$, with linear computational complexity $\mathcal{O}\left(ND\right)$. **Bottom:** CMA-ES iterates on full covariance matrices $\boldsymbol{\Sigma}_{\text{CMA}}$ with quadratic computational complexity $\mathcal{O}\left(N^2D^2\right)$.

(except for the left-hand plot) indicates the minimum cost measured in any experiment. Note also the jump in all the cost profiles from $\approx 10^3 - 10^4$ to $\approx 10^0 - 10^1$, which reflects if the updated solution is collision-free. We observe that the choice of CMA-ES vs. sep-CMA-ES does not have a substantial impact on the cost evolution for this particular problem, indicating that it is justified to use sep-CMA-ES with linear complexity. However, we observe a substantial impact when using the presented Cholesky factorization, imposing smoothness on the candidate trajectories. In all experiments using the Cholesky factorization, it converged to a collision-free solution after 3 iterations at maximum. This is an especially important result justifying the use of the Cholesky factorization inside the MPC loop, as the real-time requirements limit the number of iterations.

3.1 Extensions to VP-STO in Subsequent Work

We have built upon the VP-STO framework in two subsequent works: *i)* to explicitly account for uncertainty in the planning process via chance-constraints (Chapter 5), and *ii)* to do belief space control for robust manipulation in Jankowski et al. (2025a). In this section, we briefly summarize the extensions that we developed in the latter work, enabling the use of informative priors in the sampling process of VP-STO, and allowing for more efficient optimisation in contact-rich tasks.

3.1.1 Discussion of Via-Point-Based Trajectory Representation

In our original work, we introduced the via-point-based trajectory representation as a means to efficiently parameterise robot trajectories for sampling-based MPC. This representation allows us to define a trajectory by a set of via-points that the robot must pass through, enabling smooth and time-optimal trajectories with a reduced number of parameters.

Recap of Via-Point-based Trajectory Representation We proposed a via-point based trajectory representation, where a trajectory is defined by a set of N via-points $\boldsymbol{q}_{1:N}$. The robot configuration at time $t \in [0, T]$ is given by

$$\boldsymbol{q}(t) = \boldsymbol{\Phi}_{\text{via}}(t)\boldsymbol{\theta} + \boldsymbol{\phi}_0(t, \boldsymbol{q}_0, \dot{\boldsymbol{q}}_0), \tag{3.1}$$

where the robot trajectory is parameterised by the vector of via-points $\boldsymbol{\theta}$, i.e.,

$$\boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{q}_{\text{via}}^1 \\ \vdots \\ \boldsymbol{q}_{\text{via}}^N \end{pmatrix} \in \mathbb{R}^{N \cdot n_{\text{dof}}^r}, \tag{3.2}$$

where n_{dof}^r is the number of degrees of freedom of the robot. The basis function matrix $\boldsymbol{\Phi}_{\text{via}}(t)$ enforces that the trajectory passes through the via-points with minimal acceleration subject to the velocity being zero at the end of the trajectory. In addition, the basis offset $\boldsymbol{\phi}_0$ incorporates the initial robot configuration and velocity at the start of the trajectory[1].

[1] We refer to Jankowski et al. (2022) for implementation details on the basis functions and offsets.

Time Parameterisation Given a set of via-points $\boldsymbol{\theta}$ and the starting configuration and velocity, VP-STO finds the optimal motion duration T via a time-scaling algorithm that uses the maximum allowed joint velocities and accelerations of the robot to compute a timing-optimal trajectory. For an in-depth description of the time-scaling algorithm, beyond Sec. 4.A in the original work, we refer the reader to Section 3.2 of Jankowski (2025).

Connection to Direct Collocation VP–STO bears a conceptual similarity to classical direct collocation methods (Hargraves and Paris, 1987), in that both approaches represent trajectories through a finite set of decision variables defined at discrete points. In direct collocation, these points define the break points for states and controls, and dynamic feasibility is enforced through collocation constraints that ensure the discretized trajectory satisfies the system dynamics over each interval. In contrast, VP–STO uses via-points to parameterise a smooth geometric trajectory that minimises acceleration by construction, without enforcing the system dynamics during geometric path generation. Instead of collocation constraints, VP–STO imposes kinodynamic feasibility through a separate time-scaling procedure that enforces joint velocity and acceleration limits over a set of evaluation points. This preserves many of the advantages of collocation, i.e., low-dimensional parameterisation, smoothness, and efficient evaluation, while avoiding the need to solve a dynamics-constrained nonlinear program. Finally, this design is motivated by the use of this parameterisation within a shooting-based optimisation framework, where dynamic feasibility is handled through forward simulation rather than through the direct transcription paradigm.

3.1.2 Incorporating Informative Priors

The efficiency of the stochastic optimisation process in VP-STO is highly dependent on the quality of samples taken from the sampling distribution. In the original VP-STO formulation, we directly sampled the via-points from an uninformed Gaussian distribution, e.g., using a straight-line trajectory as the mean and a

diagonal covariance matrix. We also describe how we can incorporate a smoothness prior into the sampling process in Section V.C of the original work. However, we had not formalised this concept in a more general manner, due to space constraints.

The approach we propose here is based on the *product of experts* framework (Hinton, 2002), which allows us to combine multiple Gaussian distributions, each representing different sources of prior knowledge about the task. The aim is to construct a Gaussian sampling distribution over the via-point parameters that incorporates prior knowledge about the task. This is is linked to the general goal of sampling trajectories that are more likely to yield low-cost solutions, which can significantly improve the efficiency of the stochastic optimisation process. Given a cost function $f_c(\boldsymbol{q}_{0:T})$ that evaluates the quality of a trajectory $\boldsymbol{q}_{0:T}$, we can define a distribution over trajectories as

$$p(\boldsymbol{q}_{0:T}) \propto \exp(-f_c(\boldsymbol{q}_{0:T})). \tag{3.3}$$

If the trajectory cost is given as a sum of multiple objectives, i.e., $c(\boldsymbol{q}_{0:T}) = \sum_i c_i(\boldsymbol{q}_{0:T})$, the corresponding probability distribution can be written as a product of multiple probability distributions, i.e., $p(\boldsymbol{q}_{0:T}) \propto \prod_i e^{-c_i(\boldsymbol{q}_{0:T})}$. While directly sampling from the distribution in Eq. (3.3) is often infeasible, we show two examples of how we can construct informative Gaussian priors, each representing different sources of prior knowledge, that we can combine as a product of Gaussian experts. This will improve the sample efficiency of the stochastic optimisation process in contrast to sampling from an uninformed distribution. We begin by providing a more detailed description of the via-point based trajectory representation used in VP-STO, before explaining the general concept of the product of Gaussian experts, and finally illustrating how we use this to incorporate a contact prior and a smoothness prior into the sampling process. We illustrate this by showing *a)* how to design a Gaussian contact prior (cf. Figure 3.1), which is particularly useful for contact-rich tasks, that require the robot to make contact with an object in order to successfully complete the task; and *b)* how to design a smoothness prior given the via-point based trajectory representation, as introduced in our original

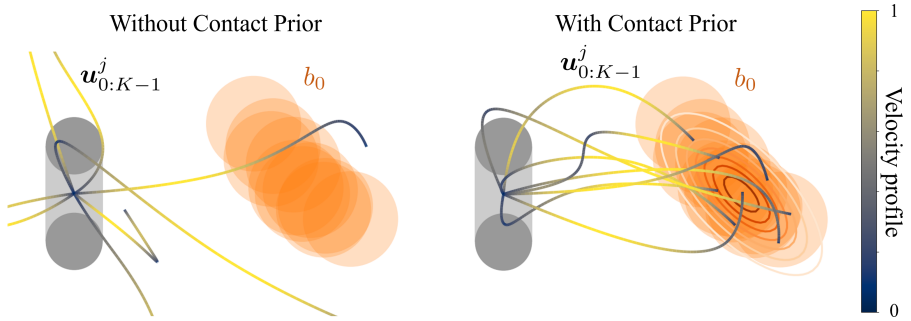


Figure 3.1: Illustration of the sampling process in VP-STO with informative priors versus uninformed sampling, taken from our work (Jankowski et al., 2025a). ***Left: Uninformed sampling***, where via-points are drawn from a simple Gaussian distribution. ***Right: Informed sampling***, where the distribution incorporates a contact prior, biasing trajectories toward regions of the state space that are more likely to establish contact with the object, given the belief over its location b_o.

work, to ensure that the sampled trajectories are smooth. In the following, we discuss how to construct informative Gaussian sampling distributions over the via-point parameters $\boldsymbol{\theta}$, which maps to a distribution over smooth, continuous and timing-optimal trajectories $\boldsymbol{q}(t)$ via the affine mapping in Eq. 3.1.

Product of Gaussian Experts We can construct a Gaussian sampling distribution over the via-point parameters $\boldsymbol{\theta}$ by combining multiple Gaussian distributions, each representing different sources of prior knowledge. The product of P Gaussian distributions $p_i(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ results in another Gaussian distribution $p(\boldsymbol{\theta})$ given by

$$p(\boldsymbol{\theta}) \propto \prod_{i=1}^{P} p_i(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta), \tag{3.4}$$

where the mean $\boldsymbol{\mu}_\theta$ and covariance $\boldsymbol{\Sigma}_\theta$ of the resulting distribution are given by

$$\boldsymbol{\Sigma}_\theta = \left(\sum_{i=1}^{P} \boldsymbol{\Sigma}_i \right)^{-1}, \tag{3.5}$$

$$\boldsymbol{\mu}_\theta = \boldsymbol{\Sigma}_\theta \left(\sum_{i=1}^{P} \boldsymbol{\Sigma}_i^{-1} \boldsymbol{\mu}_i \right). \tag{3.6}$$

Informed Sampling with Gaussian Priors We can now use the product of Gaussian experts framework to construct a *prior* Gaussian distribution over the via-point parameters $\boldsymbol{\theta}$, i.e., $p_{\text{prior}}(\boldsymbol{\theta})$, encoding a probabilistic initial guess. Compared

to a naive uninformed initial sampling distribution, e.g., white noise with a scaled variance, the prior distribution already shifts the probability mass toward regions of the state space that are more likely to yield low-cost solutions. We use CMA-ES (Hansen and Ostermeier, 2001) to iteratively update a Gaussian sampling distribution. Yet, instead of defining the CMA-ES search distribution directly over the via-point parameters $\boldsymbol{\theta}$, we define it over a latent variable $\boldsymbol{\epsilon} \in \mathcal{R}^{N \cdot n^r_{\text{dof}}}$, which is then mapped to the via-point parameters $\boldsymbol{\theta}$ via the prior Gaussian distribution, i.e.,

$$\boldsymbol{\theta} = \boldsymbol{\mu}_{\text{prior}} + \boldsymbol{L}_{\text{prior}} \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\mu}_{\text{CMA}}, \boldsymbol{\Sigma}_{\text{CMA}}), \tag{3.7}$$

where $\boldsymbol{L}_{\text{prior}}$ is the Cholesky decomposition of the prior covariance matrix $\boldsymbol{\Sigma}_{\text{prior}}$, and $\boldsymbol{\mu}_{\text{prior}}$ is the prior mean. This reparametrisation decouples the optimisation variable $\boldsymbol{\epsilon}$ from the prior distribution. At each VP-STO iteration, we then draw M candidate trajectories according to

$$\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}_{\text{prior}} + \boldsymbol{L}_{\text{prior}} \boldsymbol{\mu}_{\text{CMA}}, \boldsymbol{L}_{\text{prior}} \boldsymbol{\Sigma}_{\text{CMA}} \boldsymbol{L}^{\top}_{\text{prior}}), \tag{3.8}$$

where $\boldsymbol{\mu}_{\text{CMA}}$ and $\boldsymbol{\Sigma}_{\text{CMA}}$ are the current mean and covariance of the CMA-ES search distribution, respectively. With CMA-ES initialized as white noise, i.e., $\boldsymbol{\mu}_{\text{CMA}} = \mathbf{0}$ and $\boldsymbol{\Sigma}_{\text{CMA}} = \boldsymbol{I}$, the resulting initial sampling distribution coincides with the prior $p_{\text{prior}}(\boldsymbol{\theta})$. Eventually, given a sampled trajectory parameter $\boldsymbol{\theta}$, we find the control trajectory by discretising the continuous robot trajectory in Eq. (3.1) according to

$$\boldsymbol{u}_k = \boldsymbol{q}^r \left(t = T \frac{k+1}{K} \right), \quad k = 0, \dots, K-1, \tag{3.9}$$

where K is the number of discretization steps and T is the optimal movement duration found via time-scaling. In the following, we illustrate how to construct two specific Gaussian priors: a contact prior and a smoothness prior.

Contact Prior The contact prior biases the sampling distribution toward robot configurations that are likely to establish contact with the object of interest, i.e., $\boldsymbol{q}^o \in \mathcal{R}^{n^o_{\text{dof}}}$, where n^o_{dof} is the number of degrees of freedom of the object configuration. In the following, we assume that the object configuration follows a Gaussian

distribution, i.e., $\boldsymbol{q}^o \sim \mathcal{N}(\boldsymbol{\mu}^o, \boldsymbol{\Sigma}^o)$. For clarity, in the following, we will denote robot configurations as $\boldsymbol{q}^r$ to distinguish them from object configurations $\boldsymbol{q}^o$. With the goal of sampling diverse manipulation trajectories that are likely to make contact with the object, we restrict the prior to act only on the final via-point $\boldsymbol{q}_{\text{via}}^N$ of Eq. (3.1), which also corresponds to the final robot configuration at the end of the trajectory, i.e., $\boldsymbol{q}(T) = \boldsymbol{q}_{\text{via}}^N$. Given a function, that maps the object configuration to a corresponding contact configuration of the robot, i.e., $\boldsymbol{q}_{\text{contact}}^r = \boldsymbol{f}_{\text{contact}}(\boldsymbol{q}^o)$, we define a distribution over robot configurations that are likely to make contact with the object. Suppose that the probability density of a robot configuration making contact with the object can be approximated as

$$p_{\text{contact}}(\boldsymbol{q}^r|\boldsymbol{q}^o) = \mathcal{N}(\boldsymbol{f}_{\text{contact}}(\boldsymbol{q}^o), \boldsymbol{\Sigma}^{r|o}). \tag{3.10}$$

Marginalizing over $\boldsymbol{q}^o$ yields a distribution over robot configurations,

$$p_{\text{contact}}(\boldsymbol{q}^r) = \mathcal{N}\Big(\boldsymbol{f}_{\text{contact}}(\boldsymbol{\mu}^o), \boldsymbol{\Sigma}^{r|o} + \boldsymbol{A}\boldsymbol{\Sigma}^o\boldsymbol{A}^\top\Big), \tag{3.11}$$

where $\boldsymbol{A} = \frac{\partial \boldsymbol{f}_{\text{contact}}}{\partial \boldsymbol{q}^o}\Big|_{\boldsymbol{\mu}^o}$. Now that we have defined a distribution over robot configurations that are likely to make contact with the object, we can use this to construct a Gaussian prior over the full trajectory via-point parameters $\boldsymbol{\theta}$. The corresponding prior is then given by

$$\begin{aligned} p_{\text{contact}}\left(\boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{q}_{\text{via}}^1 \\ \vdots \\ \boldsymbol{q}_{\text{via}}^N \end{pmatrix}\right) &= \mathcal{N}\left(\begin{pmatrix} \boldsymbol{0} \\ \vdots \\ \bar{\boldsymbol{q}}_{\text{contact}} \end{pmatrix}, \begin{pmatrix} \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{Q}_{\boldsymbol{q}} \end{pmatrix}^{-1}\right) \\ &= \mathcal{N}(\bar{\boldsymbol{\theta}}_{\text{contact}}, \boldsymbol{Q}_{\boldsymbol{\theta}}^{-1}) \end{aligned} \tag{3.12}$$

where $\boldsymbol{Q}_{\boldsymbol{q}} = \left(\boldsymbol{\Sigma}^{r|o} + \boldsymbol{A}\boldsymbol{\Sigma}^o\boldsymbol{A}^\top\right)^{-1}$ describes the precision matrix of the contact prior with respect to the mean contact configuration $\bar{\boldsymbol{q}}_{\text{contact}} = \boldsymbol{f}_{\text{contact}}(\boldsymbol{\mu}^o)$.

However, the resulting precision matrix $\boldsymbol{Q}_{\boldsymbol{\theta}}^{-1}$ is degenerate in its current form, as it only sets the precision for the final via-point, while the other via-points have infinite variance. We solve this issue by combining the contact prior with the smoothness prior described below.

Contact Prior in Joint Space While the above provides a general formulation for the contact prior over robot configurations, we also want to bias the prior such that contacts are established specifically at the end-effector. To this end, we first define the contact prior in the *task space* of the robot end-effector. Let the forward kinematics $\boldsymbol{f}_{\text{fk}} : \mathbb{R}^{n^r_{\text{dof}}} \to \mathbb{R}^{n^r_{\text{task}}}$ map robot configurations $\boldsymbol{q}^r$ to end-effector poses $\boldsymbol{x}^r \in \mathbb{R}^{n^r_{\text{task}}}$, where n^r_{task} is the dimensionality of the task space (e.g., $n^r_{\text{task}} = 3$ for 3D position control). We also assume a function $\hat{\boldsymbol{f}}_{\text{contact}}$ that maps object configurations $\boldsymbol{q}^o$ to end-effector contact poses, i.e.,

$$\boldsymbol{x}^r_{\text{contact}} = \hat{\boldsymbol{f}}_{\text{contact}}(\boldsymbol{q}^o). \tag{3.13}$$

Using this, the contact prior in task space becomes

$$\hat{p}_{\text{contact}}(\boldsymbol{x}^r) = \mathcal{N}\Big(\hat{\boldsymbol{f}}_{\text{contact}}(\boldsymbol{\mu}^o),\ \boldsymbol{\Sigma}^{r|o} + \boldsymbol{A}\boldsymbol{\Sigma}^o\boldsymbol{A}^\top\Big). \tag{3.14}$$

Since we ultimately want to sample in *joint space*, we need to transform this task-space prior into a joint-space distribution. However, the forward kinematics $\boldsymbol{x}^r = \boldsymbol{f}_{\text{fk}}(\boldsymbol{q}^r)$ are nonlinear, which means that a Gaussian in task space does not map exactly to a Gaussian in joint space. To preserve tractability, we approximate the mapping locally using a first-order Taylor expansion around the mean contact configuration $\bar{\boldsymbol{q}}^r_{\text{contact}}$, yielding

$$\boldsymbol{x}^r \ \approx\ \boldsymbol{f}_{\text{fk}}(\bar{\boldsymbol{q}}^r_{\text{contact}}) + \boldsymbol{J}(\bar{\boldsymbol{q}}^r_{\text{contact}})(\boldsymbol{q}^r - \bar{\boldsymbol{q}}^r_{\text{contact}}), \tag{3.15}$$

where $\boldsymbol{J}(\boldsymbol{q}) = {}^{\partial \boldsymbol{x}^r}/_{\partial \boldsymbol{q}^r}\Big|_{\bar{\boldsymbol{q}}^r_{\text{contact}}}$ denotes the Jacobian with respect to the end-effector configuration. The mean joint configuration $\bar{\boldsymbol{q}}^r_{\text{contact}}$ can be obtained by solving the inverse kinematics problem, i.e.,

$$\bar{\boldsymbol{q}}^r_{\text{contact}} = \boldsymbol{f}_{\text{ik}}\Big(\hat{\boldsymbol{f}}_{\text{contact}}(\boldsymbol{\mu}^o)\Big). \tag{3.16}$$

This linearisation allows us to approximate the task-space Gaussian as a Gaussian in joint space, with the corresponding precision matrix

$$\hat{\boldsymbol{Q}}_{\boldsymbol{q}} = \boldsymbol{J}(\bar{\boldsymbol{q}}^r_{\text{contact}})^\top\, \boldsymbol{Q}^{-1}\, \boldsymbol{J}(\bar{\boldsymbol{q}}^r_{\text{contact}}). \tag{3.17}$$

Thus, we obtain a joint-space contact prior as in Eq. (3.12).

Smoothness Prior We incorporate temporal correlations between via-points by defining a Gaussian prior that assigns high likelihood to trajectories with low accelerations. A common approach to enforce such smoothness in trajectory optimisation is to minimise the integral over squared accelerations,

$$J_{\text{smooth}} = \tfrac{1}{2} \int_0^T \ddot{\boldsymbol{q}}^{r\top}(t)\, \boldsymbol{R}_{\boldsymbol{q}}\, \ddot{\boldsymbol{q}}^{r}(t)\, dt, \tag{3.18}$$

where $\boldsymbol{R}_{\boldsymbol{q}} \succ 0$ specifies per-joint smoothing weights. Using the trajectory parameterisation in Eq. (3.1), the acceleration can be expressed as an affine function of the via-point parameters $\boldsymbol{\theta}$ and the boundary conditions (initial configuration $\boldsymbol{q}_0^r$ and velocity $\dot{\boldsymbol{q}}_0^r$). More compactly, we write

$$\ddot{\boldsymbol{q}}^{r}(t) = \ddot{\boldsymbol{\Phi}}(t)\, \boldsymbol{w}, \tag{3.19}$$

with trajectory parameter $\boldsymbol{w} = [\boldsymbol{\theta}^\top, \boldsymbol{w}_{\text{bc}}]^\top$, where $\boldsymbol{w}_{\text{bc}} = [\boldsymbol{q}_0^r, \dot{\boldsymbol{q}}_0^r]$, and the basis function matrix $\ddot{\boldsymbol{\Phi}}(t)$ given by

$$\ddot{\boldsymbol{\Phi}}(t) = \begin{pmatrix} \ddot{\boldsymbol{\Phi}}_{\text{via}}(t) & \ddot{\boldsymbol{\Phi}}_{\text{bc}}(t) \end{pmatrix}. \tag{3.20}$$

Substituting into J_{smooth} yields

$$J_{\text{smooth}} = \tfrac{1}{2} \boldsymbol{w}^\top \boldsymbol{R}_{\boldsymbol{w}} \boldsymbol{w}, \qquad \boldsymbol{R}_{\boldsymbol{w}} = \int_0^T \ddot{\boldsymbol{\Phi}}^\top(t)\, \boldsymbol{R}_{\boldsymbol{q}}\, \ddot{\boldsymbol{\Phi}}(t)\, dt. \tag{3.21}$$

The solution of the integral yields a smoothness matrix $\boldsymbol{R}_{\boldsymbol{w}}$ that can be written in block form as

$$\boldsymbol{R}_{\boldsymbol{w}} = \begin{pmatrix} \boldsymbol{R}_{\boldsymbol{\theta}} & \boldsymbol{R}_{\boldsymbol{\theta}|\boldsymbol{w}_{\text{bc}}} \\ \boldsymbol{R}_{\boldsymbol{\theta}|\boldsymbol{w}_{\text{bc}}}^\top & \boldsymbol{R}_{\boldsymbol{w}_{\text{bc}}} \end{pmatrix}. \tag{3.22}$$

With this result, the smoothness objective is in fact a quadratic function in $\boldsymbol{w}$. Exponentiating the negative cost turns this smoothness objective into a Gaussian distribution over $\boldsymbol{w}$,

$$p_{\text{smooth}}(\boldsymbol{w}) \propto \exp(-J_{\text{smooth}}) = \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{R}_{\boldsymbol{w}}^{-1}\right). \tag{3.23}$$

This joint Gaussian couples the via-point parameters $\boldsymbol{\theta}$ with the boundary conditions $\boldsymbol{w}_{\text{bc}}$. Since the boundary conditions are known when constructing the prior, we condition on $\boldsymbol{w}_{\text{bc}}$ to obtain a Gaussian distribution over the via-points:

$$p_{\text{smooth}}(\boldsymbol{\theta} \mid \boldsymbol{q}_0^r, \dot{\boldsymbol{q}}_0^r) = \mathcal{N}\left(\bar{\boldsymbol{\theta}}_{\text{smooth}}, \boldsymbol{R}_{\boldsymbol{\theta}}^{-1}\right), \tag{3.24}$$

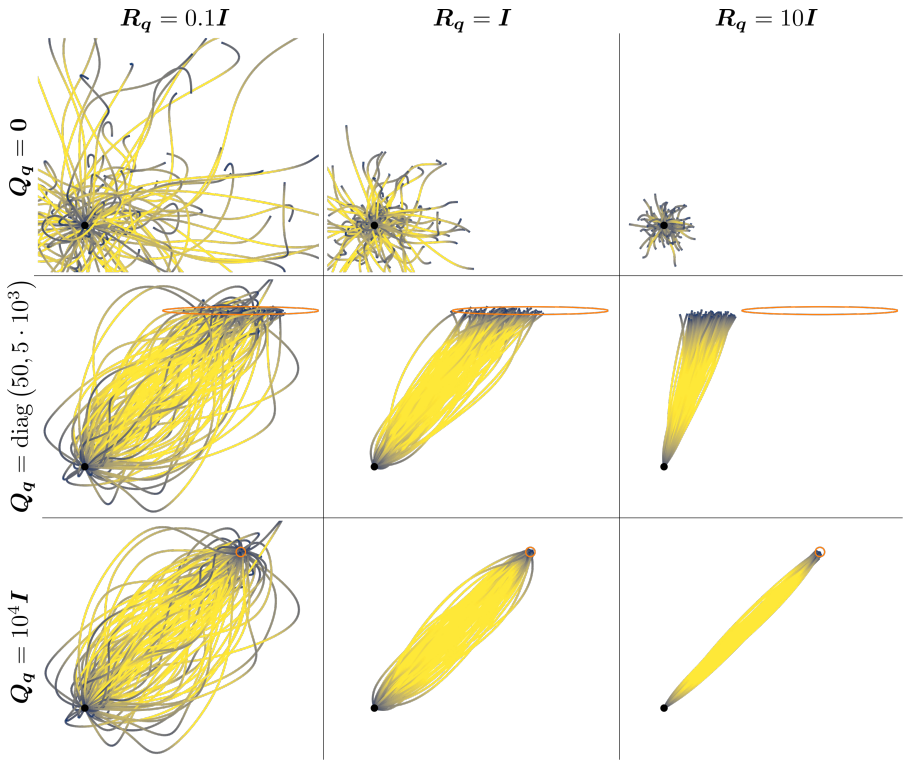


Figure 3.2: ***Smooth trajectories sampled from the product of the smoothness and contact priors.*** The contact prior is shown as orange ellipses and circles, with its mean in the center. The respective velocity profiles of the trajectories are encoded through color with low velocities in blue and high velocities in yellow. All trajectories begin and end with zero velocity. Each sub-figure shows different combinations of the precision matrices $\boldsymbol{Q}_q$ and $\boldsymbol{R}_q$, which control the influence of the contact prior and the smoothness prior, respectively.

where the precision matrix is given by

$$\boldsymbol{R}_{\boldsymbol{\theta}} = \int_0^T \ddot{\boldsymbol{\Phi}}_{\text{via}}^{\top}(t)\, \boldsymbol{R}_{\boldsymbol{q}}\, \ddot{\boldsymbol{\Phi}}_{\text{via}}(t)\, dt, \tag{3.25}$$

and the prior mean

$$\bar{\boldsymbol{\theta}}_{\text{smooth}} = \boldsymbol{R}_{\boldsymbol{\theta}}^{-1} \boldsymbol{R}_{\boldsymbol{\theta}|\boldsymbol{w}_{\text{bc}}}, \boldsymbol{w}_{\text{bc}} \tag{3.26}$$

captures the dependence on the boundary conditions. Importantly, the smoothness prior precision matrix $\boldsymbol{R}_{\boldsymbol{\theta}}$ depends only on the chosen basis functions and smoothing weights, and can therefore be precomputed offline.

Combining Contact and Smoothness Prior We can now combine the contact prior $p_{\text{contact}}(\boldsymbol{\theta})$ from Eq. (3.12) with the smoothness prior $p_{\text{smooth}}(\boldsymbol{\theta} \mid \boldsymbol{q}_0^r, \dot{\boldsymbol{q}}_0^r)$ to

form a product of experts, as described in Eq. (3.4). Since the product of two multivariate Gaussians is again a Gaussian, the resulting prior distribution is a multivariate Gaussian with parameters

$$\boldsymbol{\Sigma}_\theta = \left(\boldsymbol{Q}_{\boldsymbol{\theta}} + \boldsymbol{R}_{\boldsymbol{\theta}}\right)^{-1}, \tag{3.27}$$

$$\boldsymbol{\mu}_\theta = \boldsymbol{\Sigma}_\theta \left(\boldsymbol{Q}_{\boldsymbol{\theta}} \bar{\boldsymbol{\theta}}_{\text{contact}} + \boldsymbol{R}_{\boldsymbol{\theta}} \bar{\boldsymbol{\theta}}_{\text{smooth}}\right). \tag{3.28}$$

Figure 3.2 illustrates samples drawn from the product of the contact prior and the smoothness prior, depending on the choice of the precision matrices $\boldsymbol{Q}_{\boldsymbol{q}}$ and $\boldsymbol{R}_{\boldsymbol{q}}$. Each sub-figure shows trajectory candidates drawn from a single Gaussian distribution over via-point parameters $\boldsymbol{\theta}$. All trajectories start and end with exactly zero velocity, as enforced by the boundary conditions.

Gaussian Priors as Regularisers in the Quasi-Newton View of CMA-ES As discussed in Hansen (2016), the objective of covariance matrix adaptation is to approximate the local contour geometry of the objective function. On convex–quadratic functions, this adaptation corresponds to learning an estimate of the inverse Hessian, thereby rendering CMA-ES analogous to a stochastic quasi-Newton method. In this view, the evolving covariance matrix provides a sample-based approximation of curvature information, updated through ranked candidate evaluations rather than explicit derivatives. Within this quasi-Newton interpretation, the Gaussian priors introduced through the product-of-experts construction naturally act as *regularisers*: their precision matrices contribute additional curvature to the effective inverse-Hessian approximation encoded by the sampling covariance. As a result, the priors bias the search toward regions of the parameter space consistent with task-specific structure (e.g., smoothness or expected contact configurations), thereby improving sample efficiency and stabilizing the overall optimisation process.

Implementation Detail on CMA-ES with Informative Priors With the use of informative priors, we can also modify the CMA-ES itself to further improve the computational efficiency of the overall VP-STO framework. Instead of using the standard CMA-ES algorithm, we can exploit the fact that we are already using

a Cholesky factorisation of the covariance matrix that, via the smoothness prior, already encodes correlations between the via-points. Thus, instead of updating the full covariance matrix $\mathbf{\Sigma}_{\text{CMA}}$ of CMA-ES, we only update its diagonal, which is known as sep-CMA-ES (Ros and Hansen, 2008). This improves the computational complexity from $\mathcal{O}(N^2D^2)$ (CMA-ES) to $\mathcal{O}(ND)$ (sep-CMA-ES), meaning that the computational load of sampling from and updating the covariance matrix scales linearly with the number of via-points N and the n^r_{dof} DoF of the robot.

3.2 Ablations

In this section, we present ablation studies that further justify some of the design choices made in the VP-STO framework.

Impact of the Number of Via-Points In this ablation study, we investigate the impact of the number of via-points used to represent the robot movement. This hyper-parameter has a high impact on the overall framework performance. On one hand, it directly sets the dimensionality of the optimisation problem to solve; on the other hand, it directly spans the space of movements that can be synthesised. From an optimisation perspective, adjusting the number of via-points provides an intuitive mechanism to trade off solution quality against computational cost. We illustrate this relationship in Fig. 3.3, where we let a 1D double-integrator move from $q_0 = 0.0$, $\dot{q}_0 = 0.0$ to $q_T = 1.0$, $\dot{q}_T = 0.0$ in minimal time, subject to a maximum velocity $|\dot{q}| < 0.1$ and an acceleration limit $|\ddot{q}| < 0.2$; with a varying number of via-points. This time-optimal control problem is known to be solved by a bang-bang acceleration profile, such that we know the analytic limit of the minimal time to be $c_{\text{bang-bang}} = T_{\text{bang-bang}} = 10.5$, which is depicted as dashed black line in the upper-left plot. We observe that the solution cost exponentially converges to $c_{\text{bang-bang}}$ as we increase the number of via-points. The lower-left plot shows the number of CMA-ES-iterations required to converge as a function of the number of via-points. Here, we detect convergence if $|c_k - c_{k-1}| < 10^{-6}$ in the k-th iteration. Interestingly, the number of iterations grows linearly with the number of via-points. Note that

this does not mean that the computational cost grows linearly with the number of via-points, since the computational cost for a single iteration is either linear (sep-CMA-ES) or quadratic (CMA-ES) in the number of via-points. Nevertheless, these results motivate the use of a small number of via-points, as the marginal benefit of adding additional via-points does not justify the increased computational cost.

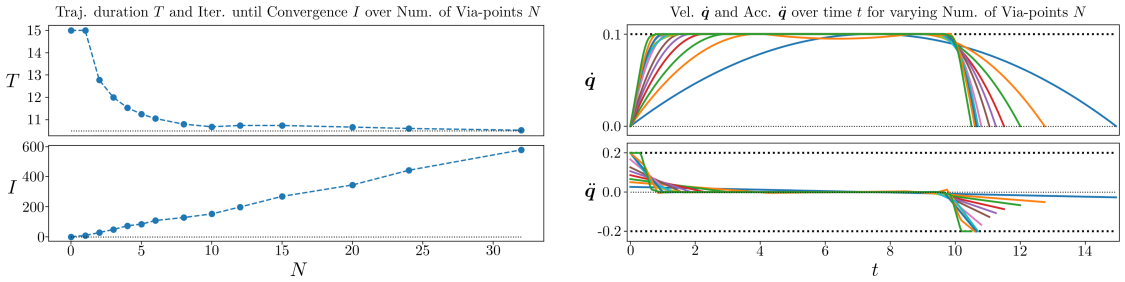


Figure 3.3: A study of the impact of the number of via-points in a 1D time-optimisation problem. **Top-Left:** Impact on the resulting movement duration. The dotted black line illustrates the duration of the optimal *bang-bang* solution. **Bottom-Left:** Impact on the number of iterations required until convergence. **Right:** Velocity and acceleration profiles for evaluated numbers of via-points.

Impact of the Cholesky Factorisation of the Covariance Matrix In this ablation study, we look at a 2D minimal-time planning problem including an obstacle that is to be avoided. We fix the number of via-points to $N = 6$ and set up four different optimisation loops that are supposed to solve the same problem. Each setup uses either CMA-ES or sep-CMA-ES and runs with or without the Cholesky factorisation of the covariance matrix as described in Sec. 3.1. For comparison, we look at the cost evolution over the number of iterations. The dashed black line in all plots (except for the left-hand plot) indicates the minimum cost measured in any experiment. Note also the jump in all the cost profiles from $\approx 10^3 - 10^4$ to $\approx 10^0 - 10^1$, which reflects if the updated solution is collision-free. We observe that the choice of CMA-ES vs. sep-CMA-ES does not have a substantial impact on the cost evolution for this particular problem, indicating that it is justified to use sep-CMA-ES with linear complexity. However, we observe a substantial impact when using the presented Cholesky factorisation, imposing smoothness on the candidate trajectories. In all experiments using the Cholesky factorisation, it

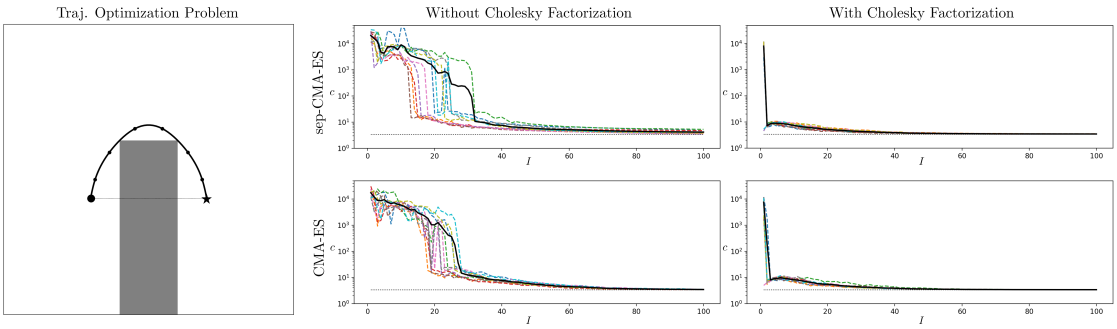


Figure 3.4: A study of the impact of the Cholesky factorisation of the Covariance Matrix $\boldsymbol{\Sigma}_{\text{via}}$ in a 2D time-optimisation problem with obstacle avoidance. **Left:** The configuration space including the obstacle in gray, the initial guess as dashed line, and the optimal solution around the obstacle as solid line together with the corresponding via-points as circles. **Center:** The via-point covariance matrix is explicitly updated, i.e., $\boldsymbol{\Sigma}_{\text{via}} = \boldsymbol{\Sigma}_{\text{CMA}}$. **Right:** The via-point covariance matrix is updated through a Cholesky factorisation, i.e., $\boldsymbol{\Sigma}_{\text{via}} = \boldsymbol{L}\boldsymbol{\Sigma}_{\text{CMA}}\boldsymbol{L}^{\top}$. **Top:** sep-CMA-ES iterates on diagonal covariance matrices only, i.e., $\boldsymbol{\Sigma}_{\text{CMA}} = \text{diag}(\boldsymbol{\sigma}_{\text{CMA}})$, with linear computational complexity $\mathcal{O}(ND)$. **Bottom:** CMA-ES iterates on full covariance matrices $\boldsymbol{\Sigma}_{\text{CMA}}$ with quadratic computational complexity $\mathcal{O}(N^2D^2)$.

converged to a collision-free solution after 3 iterations at maximum. This is an especially important result justifying the use of the Cholesky factorisation inside the MPC loop, as the real-time requirements limit the number of iterations.

3.3 Limitations and Future Work

Holonomic Systems Only So far, we have assumed holonomic robot dynamics, i.e., the robot can move in any direction in its configuration space. This assumption holds for robotic manipulators, but not for mobile robots or legged robots. While we explored several strategies to apply VP-STO to non-holonomic systems in the past, we did not validate them sufficiently. Future work could extend these strategies and perform more rigorous evaluations on non-holonomic systems.

Computational Efficiency Given that VP-STO is a shooting method (cf. Chapter 2.3.1), its real-time performance depends on the computational efficiency of the underlying dynamics model. In the original work, we used a quasi-static dynamics model for the non-prehensile pushing task, which is computationally efficient but does not extend to more complex tasks, such as tasks that involve more dynamic

settings and multiple possible contacts with different parts of the robot. The use of more accurate physics engines, such as MuJoCo, allows to extend VP-STO to more complex tasks, but comes at the cost of increased computation time, requiring fewer samples and a shorter time horizon to achieve real-time performance. This limitation has motivated the work in the subsequent Chapter 4, where we learn a generative model of the dynamics to inform the sampling process and improve the efficiency of sampling-based MPC. While Chapter 4 focuses on improving the efficiency of sampling-based MPC via learned generative models, future work could explore the combination of these generative models with VP-STO to further improve its efficiency and real-time performance.

Beyond Unimodal Gaussian Distributions VP-STO in its current form uses a unimodal Gaussian distribution to sample via-point parameters that map to robot trajectories. This choice is motivated by the ability to incorporate informative priors via the product of Gaussians, as described above, as well as by the use of CMA-ES, which assumes a Gaussian search distribution. However, in some tasks, the optimal solution may be multi-modal, meaning that there are multiple distinct trajectories that can achieve the task successfully. In such cases, a unimodal Gaussian distribution may not be sufficient to capture the diversity of possible solutions. While replanning at high frequencies can help to overcome this limitation, future work could explore the use of other sampling distributions, that are better suited to capture multi-modal solutions. A different approach to address this limitation could be to use multiple VP-STO instances in parallel, each with a different initial guess or prior, akin to a multi-population evolutionary strategy (Branke et al., 2000). This could allow the system to explore multiple modes of the solution space simultaneously, increasing the likelihood of finding a successful trajectory.

State Estimation Like in all model-based planning and control approaches, reliable state estimation is crucial for the success of VP-STO. All experiments on real hardware involved a motion capture system to provide accurate state feedback. Future work should explore the integration of onboard sensors, such as cameras

or tactile sensors, to enable state estimation without external tracking systems. While this introduces uncertainty into the overall system, the VP-STO framework could be extended to account for this uncertainty, as demonstrated in Chapter 5, as well as in Chapter 6. In addition, future work could further explore the use of model-based methods in latent spaces, as done in more recent reinforcement learning approaches (Hansen et al., 2023; Wang et al., 2025b).

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	VP-STO: Via-point-based Stochastic Trajectory Optimization for Reactive Robot Behavior	
Publication Status	☑ Published	□ Accepted for Publication
	□ Submitted for Publication	□ Unpublished and unsubmitted work written in a manuscript style
Publication Details	Jankowski, J., Brudermüller, L., Hawes, N., and Calinon, S. (2023). VP-STO: Via-point-based stochastic trajectory optimization for reactive robot behavior. In 2023 IEEE International Conference on Robotics and Automation (ICRA), pages 10125–1013	

Student Confirmation

Student Name:	Lara Brudermüller		
Contribution to the Paper	– The paper was jointly developed and written in equal contributorship with Julius Jankowski. – I designed and carried out the real-world experiments, while Julius conducted the simulation experiments. – I mostly wrote the introduction and related work sections, while Julius wrote the preliminaries section on trajectory representation, as this was based on his previous work. All remaining sections were written jointly.		
Signature		Date	September 17, 2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Professor Nick Hawes		
Supervisor comments I confirm that Lara made a substantial contribution to this publication, and that the description above is accurate.			
Signature		Date	September 17, 2025

This completed form should be included in the thesis, at the end of the relevant chapter.

4

Learning to Improve Reactivity of Sampling-Based Model Predictive Control

Publication Note

This chapter presents the work from the following submission:

> Brudermüller, L., Hung, B., Zhu, X., Wang, J., Hawes, N., Culbertson, P., and Le Cleac'h, S. (2025b). Generative models from and for sampling-based mpc: A bootstrapped approach for adaptive contact-rich manipulation. Manuscript under review at *IEEE Robotics and Automation Letters*

Supplementary Material

An accompanying video is available at: `https://youtu.be/lKCGjjddv1E`.

In this chapter, we investigate how *learning* can be used to improve the *reactivity* of sampling-based Model Predictive Control (SPC). While the methods presented in Chapter 3 demonstrate that SPC can implicitly handle uncertainty through frequent replanning, their effectiveness is limited in scenarios where forward simulation is computationally expensive, such as contact-rich tasks requiring high-fidelity physics simulations. In such cases, the limited simulation budget constrains exploration during online replanning, thereby reducing the controller's reactivity. Here, we extend this line of work by exploring how data-driven components can be integrated into sampling-based MPC to accelerate online planning and enhance reactivity, while still operating in a *deterministic* environment with *no prior knowledge of*

	Chapter 3	Chapter 4	Chapter 5	Chapter 6
Theme	Reactivity	Learning for Reactivity	Robustness	Exploration
Uncertainty Handling	Implicit	Implicit	Explicit	Explicit
Prior Knowledge of Uncertainty	None	None	Sampling-based access	Sampling-based access
Environment	Deterministic	Deterministic	Stochastic	Stochastic
Method	MPC	MPC	MPC	Belief-space control
Control Problem	Min. cost	Min. cost	Min. cost s.t. chance constraints	Max. information gain

Table 1.2: Theme, settings and methods for each chapter.

uncertainty. This setting is also summarised in Table 1.2, repeated above.

The central idea is to leverage generative modelling not to replace the MPC optimisation loop, but to make it more efficient by shaping the sampling process. Specifically, we introduce a *Generative Predictive Control* (GPC) framework, which bootstraps sampling-based MPC with proposal distributions learned from open-loop control sequences collected offline. This setting allows us to preserve the strengths of MPC, adaptivity and robustness through online replanning, while addressing one of its key limitations: the inefficiency of sampling in high-dimensional action spaces. The learned proposals provide more informative samples during online optimisation, improving the sample efficiency and thereby the reactivity of the controller, as it can find high-quality solutions faster within the limited time budget available for online replanning. We note that this approach is fundamentally different from learning a control policy directly from data, as done in imitation learning or offline reinforcement learning. Instead of replacing the MPC loop with a learned policy, we retain the online optimisation process and use learning to enhance it. A distinctive aspect of this work is that, unlike approaches that rely on human demonstrations, we instead learn from planner-generated data. In particular, the proposal distributions are trained on locally-optimal yet noisy trajectories produced

by sampling-based MPC itself, making the framework self-bootstrapping while still retaining the adaptivity of online optimisation. Moreover, since the offline data is generated with larger sample sizes and longer horizons than are feasible online, the learned proposal distributions implicitly encode longer-term dependencies, enabling effective planning with shorter horizons during online execution. This works even for tasks that inherently require long-horizon reasoning. We demonstrate these benefits on a challenging contact-rich loco-manipulation task, specifically designed to require long-horizon planning in order to avoid getting trapped in local minima. Implementation details are provided in the Appendix of the paper manuscript, which is included at the end of this thesis as Appendix A.

Generative Models From and For Sampling-Based MPC: A Bootstrapped Approach For Adaptive Contact-Rich Manipulation

Lara Brudermüller[1,2,*], Brandon Hung[2], Xinghao Zhu[2], Jiuguang Wang[2], Nick Hawes[1], Preston Culbertson[2,3,†], Simon Le Cleac'h[2,†]

***Abstract*— We present a generative predictive control (GPC) framework that amortizes sampling-based Model Predictive Control (SPC) by bootstrapping it with conditional flow-matching models trained on SPC control sequences collected in simulation. Unlike prior work relying on iterative refinement or gradient-based solvers, we show that meaningful proposal distributions can be learned directly from noisy SPC data, enabling more efficient and informed sampling during online planning. We further demonstrate, for the first time, the application of this approach to real-world contact-rich loco-manipulation with a quadruped robot. Extensive experiments in simulation and on hardware show that our method improves sample efficiency, reduces planning horizon requirements, and generalizes robustly across task variations.**

I. Introduction

Reactive contact-rich (loco-)manipulation in high-dimensional state and action spaces poses significant challenges for real-time control. Sampling-based Model Predictive Control (SPC) offers a principled framework to address these challenges by solving trajectory optimization problems online with a model in the loop, enabling adaptive behavior and constraint satisfaction [1]–[4]. However, the computational cost of forward simulation, combined with the challenge of effectively exploring the search space in high-dimensional, contact-rich environments, limits the applicability of real-time optimization to more complex behaviors and higher-frequency control.

A promising line of work seeks to amortize the computational burden of online optimization by shifting it to an offline phase [5]–[7]. The key idea is to collect high quality data and train a generative model to capture a distribution of useful actions or control sequences. At test time, this model can be used to guide or warmstart the sampling distribution. The method attempts to drastically improve solution quality and efficiency by focusing sampling on high-likelihood, constraint-satisfying regions of the action space.

Recent advances in generative modeling, particularly diffusion and flow-matching models, have shown strong performance in learning expressive end-to-end policies for dexterous manipulation tasks [8]–[10]. Offline model-based reinforcement learning methods [11], [12] also show strong performance approximating optimal solutions by leveraging precomputed data to enable fast runtime control via policy networks. However, these methods are often limited by the scope of their immense training data and struggle to generalize to out-of-distribution (OOD) states or tasks. In response to these limitations, several recent works demonstrate that bootstrapping online planners with offline-trained generative models leads to faster convergence, better exploration, and more robust performance in complex environments [7], [13]–[15]. In this paper, we focus on how offline data collection and generative modeling can both accelerate and guide online sampling-based MPC in contact-rich, high-dimensional settings while maintaining the flexibility and adaptability of online optimization.

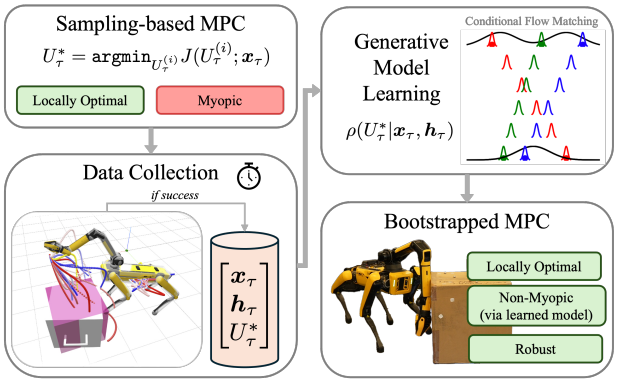


Fig. 1. *Generative predictive control (GPC) framework for bootstrapping sampling-based Model Predictive Control (SPC).* We collect open-loop control sequences from an SPC algorithm in simulation and use them to train a generative proposal distribution. At test time, this model guides and amortizes online MPC, enabling non-myopic, constraint-satisfying behavior with improved sample efficiency and robustness in contact-rich, high-dimensional settings.

Contributions: We propose a *generative predictive control (GPC) framework* that *bootstraps* SPC with conditional flow-matching models trained on SPC control sequences collected in simulation. To the best of our knowledge, we are the first to show that meaningful *proposal distributions can be learned directly from noisy SPC data* without requiring expert refinement or numerical solvers. We are also the first to demonstrate that this approach improves sample efficiency and *generalizes robustly to task variations* in both simulation and*real hardware in a contact-rich loco-manipulation task.*

II. Related Work

SPC for Contact-Rich Manipulation: SPC has been widely adopted for its robustness to nonconvex and discontinuous problems, particularly in contact-rich robotic tasks [1]–[3], [16]–[19]. These methods typically optimize a tra-

[1]Oxford Robotics Institute, University of Oxford, UK.
[2]Robotics and AI Institute (RAI), Boston, USA.
[3]Cornell University, Ithaca, NY, USA.
*This work was conducted while Lara Brudermüller was an intern at the Robotics and AI Institute.
†Preston Culbertson and Simon Le Cleac'h advised this work equally.

jectory distribution by iteratively sampling candidate controls and selecting actions based on forward-simulated costs. Their performance is often limited by the computational cost of forward dynamics, especially when simulating contact interactions or systems with many degrees of freedom (DOF). While prior work has sought to speed up forward simulation, e.g., via quasi-static approximations [20] or learned dynamics models [21], these approaches often trade off fidelity or depend on highly accurate model learning. In contrast, we do not aim to replace the dynamics model but rather to amortize the trajectory optimization itself. We do this by learning generative models over control distributions derived from open-loop control sequences that either led to task success or incurred low cost in offline sampling-based MPC, enabling faster and more informed sampling at test time.

Amortizing Online Optimization via Offline Learning: Recent work leverages offline data to reduce the computational burden of control algorithms at runtime. Common approaches include behavior cloning on expert demonstrations and planner rollouts [8]–[10] or model-based RL [11], [12]. When sourced from high-quality planners (such as sampling-based MPC), this data enables training generative models that can guide or initialize online control. Several works have explored learning from planner-generated trajectories to approximate optimal solutions and accelerate planning [6], [22]–[24]. This idea underpins Approximate MPC (AMPC), where learned models bootstrap or replace expensive solvers [25]. A representative example [26] uses diffusion models to approximate near-globally optimal MPC solutions from locally optimal trajectories generated by a numerical solver [27]. Yet, the learned models are not used to guide sampling but rather to replace the solver entirely. Our work is most closely related to [14], which also bootstraps SPC using generative models trained on SPC control sequences. Their method alternates online data collection with model updates, but we find this iterative refinement can collapse the multi-modal control distribution important for effective sampling. In contrast, our method trains a generative model over *offline* data (noisy SPC rollouts) and achieves strong performance, showing that bootstrapped SPC does not require iterative refinement. To the best of our knowledge, both [14] and our approach are the first to learn from SPC data rather than trajectories from gradient-based solvers.

Learning Sampling Distributions for Online MPC: Another line of work aims to improve SPC by learning structured priors over control sequences in latent action spaces using generative models [28], [29]. These methods typically rely on expert demonstrations, which lack exploratory diversity and require complex bi-level training to learn both the latent spaces and their distributions. In contrast, we focus on directly leveraging the diverse data produced by sampling-based MPC during offline data collection. Freed from real-time constraints, we can instead expand the search space during planning to yield richer control sequences that support efficient sampling through simpler generative models.

Infinite-Horizon Value Approximation and MPC: A complementary line of research seeks to reduce the myopia of finite-horizon MPC by learning *infinite-horizon value functions* and integrating them into the control loop [30]–[33]. These methods approximate an infinite-horizon value signal over states, which is then used as a terminal cost or shaping function for MPC, thereby injecting long-horizon foresight into an otherwise short-horizon optimizer. These approaches share the same motivation as ours, i.e., mitigating the short-horizon bias of online optimization, but are orthogonal in how they inject long-term structure. Even with an accurate estimate of the infinite-horizon value function, MPC still requires an effective mechanism for *searching* the control space. In contrast, our method focuses directly on improving this search by learning generative models over successful control sequences, thereby guiding the sampling distribution used by SPC.

III. Background

A. Problem Formulation

In this paper we consider optimal control problems in continuous action spaces. Given an initial state $\boldsymbol{x}_0 = \boldsymbol{x}_{init}$, the objective is to determine a sequence of open-loop control actions $U_\tau = [\boldsymbol{u}_\tau, \boldsymbol{u}_{\tau+1}, \dots, \boldsymbol{u}_{\tau+T}]$ that minimizes a given cost function $\ell(\boldsymbol{x}, \boldsymbol{u})$ over a finite time horizon T:

$$\min_{\boldsymbol{u}_0, \boldsymbol{u}_1, \dots, \boldsymbol{u}_T} \quad L_f(\boldsymbol{x}_{T+1}) + \sum_{\tau=0}^{T} \ell(\boldsymbol{x}_\tau, \boldsymbol{u}_\tau) \tag{1a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{\tau+1} = f(\boldsymbol{x}_\tau, \boldsymbol{u}_\tau), \quad \tau = 0, \dots, T \tag{1b}$$

$$\boldsymbol{x}_0 = \boldsymbol{x}_{\text{init}}, \tag{1c}$$

where $\boldsymbol{x}_\tau \in \mathbb{R}^n$ and $\boldsymbol{u}_\tau \in \mathbb{R}^m$ are the state vector and the control input at time step τ, respectively. The functions $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and $L_f : \mathbb{R}^n \to \mathbb{R}$ represent the stage and terminal cost, respectively. We assume access to a simulator (e.g. MuJoCo [34]) or a learned model to approximate the system dynamics $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. For a more compact notation, we define a cost function $J : \mathbb{R}^{m \times T} \times \mathbb{R}^n \to \mathbb{R}$ that encapsulates both, costs and system dynamics, allowing us to write the problem as

$$\min_{U} \quad J(U; \boldsymbol{x}_{\text{init}}). \tag{2}$$

Rather than deriving a single, globally optimal policy, MPC re-optimizes a local policy at each time step by simulating the system dynamics over a shorter receding horizon $H < T$.

B. Sampling-based MPC (SPC)

Contact-rich robot control tasks pose significant challenges due to non-convex cost functions and the nonlinear, often discontinuous nature of system dynamics. Sampling-based MPC addresses these issues by optimizing over a parameterized distribution $\pi_\phi(U)$ rather than directly computing the optimal control sequence. We consider a generic SPC procedure in which, at each control step τ, the controller samples N control sequences $\{U^{(i)}\}_{i=1}^N$ from the current distribution π_ϕ, simulates their outcomes from the current state estimate $\hat{\boldsymbol{x}}_\tau$, and evaluates them using the cost function $J(U^{(i)}; \hat{\boldsymbol{x}}_\tau)$. Based on these evaluations, the distribution

parameters ϕ are updated according to the chosen SPC algorithm. The executed control $\boldsymbol{u}_\tau$ is typically the first element of the sampled control sequence U_τ or derived via spline interpolation across the optimized sequence. We focus on diagonal Gaussian distributions of the form $\pi_\phi(U) = \mathcal{N}(\bar{U}, \Sigma)$, as used in the Cross-Entropy Method (CEM) [35] and other SPC algorithms [2], [16]. Here, $\phi = (\bar{U}, \Sigma)$, and $\Sigma = \mathrm{diag}(\boldsymbol{s})$, with $\boldsymbol{s}$ denoting a vector of variances.

C. Generative Modeling: Flow-Matching

While the above focuses on shaping a sampling distribution for SPC, generative modeling focuses on a different problem: produce a sample z from a target distribution $p(z)$, which is typically unknown in closed form, but can be approximated by a dataset of samples $\mathcal{D}$. Among recent approaches to generative modeling, two closely related approaches have gained significant traction due to their ability to capture complex, multi-modal distributions: flow matching [36] and diffusion models [37]. The underlying concept is to learn a distribution over trajectories or transformations that maps a simple prior distribution to complex target data. In this work, we focus on flow-matching models and their conditional variant [38], as they offer superior inference speed compared to diffusion models.

Flow matching aims to learn a time-dependent vector field $v_\theta(z, t)$ that transports samples from an easy-to-sample prior distribution $p_0(z)$ (e.g., a standard Gaussian) to a target data distribution $p_1(z)$. The flow network v_θ is trained on

$$\min_\theta \mathbb{E}_{\substack{z_0 \sim p_0,\\ z_1 \sim p_1,\\ t \sim \mathcal{U}(0,1)}} \left[\| v_\theta\left(t z_1 + (1-t) z_0,\, t\right) - (z_1 - z_0) \|^2 \right]$$

This loss encourages the vector field to push samples along straight-line paths from z_0 to z_1. At inference time, new samples are generated by drawing $z_0 \sim p_0(z)$ and solving the ODE $\dot{z} = v_\theta(z, t)$ from $t = 0$ to $t = 1$, typically approximated with an Euler scheme $z_{t+\delta t} = z_t + v_\theta(z_t, t)\delta t$.

IV. Bootstrapping Sampling-based MPC with Generative Flow-Matching Models

In this section, we introduce our approach to bootstrapping SPC with generative models trained on open-loop control sequences collected from SPC itself. The core idea is to learn a generative model that approximates the distribution of successful control sequences conditioned on the current state and history. At test time, this model serves as a proposal distribution to guide and warm-start the sampling process in online SPC, improving sample efficiency and robustness.

A. Data collection

The offline data collection phase is central to our approach. It should provide control sequences (conditioned on the current state and history of states) likely to lead to task success. Since we aim to bootstrap SPC at test time, we generate this dataset directly from an SPC algorithm — specifically CEM, as it is widely used and easy to tune for different tasks without the runtime constraints of online control. This allows for longer horizons and larger sample sizes to collect high-quality, non-myopic control sequences.

Given a task and associated cost function, we run CEM across multiple episodes, each with random state initializations and capped at a maximum number of MPC iterations. During each episode, we record $(\boldsymbol{x}_\tau, \boldsymbol{h}_\tau, U^*_\tau)$, where $\boldsymbol{x}_\tau$ is the current state, $\boldsymbol{h}_\tau$ encodes a fixed-window history of states/observations, and U^*_τ denotes the mean control sequence of the CEM distribution at time τ. An experiment is considered successful if the task is completed within the allowed time steps. We define $\mathcal{I}_{\text{success}}$ as the index set of all successful experiment episodes and construct our training dataset as

$$\mathcal{D} = \bigcup_{i \in \mathcal{I}_{\text{success}}} \{(\boldsymbol{x}^{(i)}_\tau, \boldsymbol{h}^{(i)}_\tau, U^{*(i)}_\tau)\}_{\tau=0}^{T_i},$$

where T_i is the final time step of episode i. This ensures that only control sequences from successful rollouts are used to train the generative model.

To reduce the complexity of the generative model while improving runtime efficiency and smoothness, each control sequence U_τ is represented using $K < H$ spline interpolation points over a planning horizon of H time steps, i.e. $U = [\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_K]$. We also employ *i)* a progress-based heuristic to reset the variances during CEM to avoid early mode collapse, and *ii)* action-level annealing [18] that increases exploration, i.e., variances, for control points further into the horizon.

B. Learning Control Sequence Proposal Distributions

Once we have collected a task dataset of open-loop trajectories, we can train a flow-matching generative model to learn a time-varying state-conditional vector field $v_\theta(U, \boldsymbol{x}_\tau, \boldsymbol{h}_\tau, t)$ that pushes samples from the noise distribution $U_{t=0} \sim \mathcal{N}(0, I)$ to the target distribution $U_{t=1} \sim p_\theta(U \mid \boldsymbol{x}_\tau, \boldsymbol{h}_\tau)$, i.e. the distribution of control sequences that are likely to lead to successful task completion given the current state $\boldsymbol{x}_\tau$ and state/observation history $\boldsymbol{h}_\tau$.

A key design choice is conditioning the model not only on the current state $\boldsymbol{x}_\tau$ but also on a short *observation history* $\boldsymbol{h}_\tau$. This provides information that is not contained in a single snapshot, such as velocities and contact transitions. Because many of our tasks are effectively non-Markovian in the observation space (due to partial observability, latency, and unmodeled dynamics), conditioning on $\boldsymbol{h}_\tau$ allows the model to infer these latent variables and produce more consistent and goal-directed control sequences. This choice is standard in robotic control and prediction models, where the observation space is often non-Markovian; e.g., [39].

For simplicity, we describe sampling from the generative model as sampling from the distribution $p_\theta(U \mid \boldsymbol{x}_\tau, \boldsymbol{h}_\tau)$. We refer to our method as generative predictive control (GPC), which leverages a learned distribution over control sequences conditioned on the task context. This distribution can be used in two distinct ways: *i)* to sample control sequences from using a random shooting approach and evaluate the best based on value functions, or *ii)* to update the sampling

Algorithm 1: GPC-CEM

Input: Current state $\boldsymbol{x}_\tau$, history $\boldsymbol{h}_\tau$
Sampling distribution $\pi_\phi(U) = \mathcal{N}(\bar{U}, \Sigma)$
Flow model $p_\theta(U \mid \boldsymbol{x}_\tau, \boldsymbol{h}_\tau)$
Number of rollouts $N = N_{\text{CEM}} + N_{\text{Flow}}$
Number of elites N_{elite}
State estimator $\hat{\boldsymbol{x}}(\tau)$

while *planning* **do**

- $\boldsymbol{x}_0 \leftarrow \hat{\boldsymbol{x}}(\tau)$ *// Get current state estimate*
- Sample N_{CEM} trajectories from CEM:
  $\{U^{(i)}\}_{i=1}^{N_{\text{CEM}}} \sim \pi_\phi(U)$
- Sample N_{Flow} trajectories from flow model:
  $\{U^{(j)}\}_{j=1}^{N_{\text{Flow}}} \sim p_\theta(U \mid \boldsymbol{x}_\tau, \boldsymbol{h}_\tau)$
- Compute $\{J^{(k)} \leftarrow J(U^{(k)}; \boldsymbol{x}_0)\}_{k=1}^{N}$ *// parallel rollouts*
- Select top N_{elite} elite trajectories with lowest cost:
  $\{U^{(k)}\}_{k=1}^{N_{\text{elite}}} \leftarrow$ elite set
- Update $\pi_\phi(U)$ using elite statistics:
  $U^* \leftarrow U^{(k^*)}, \quad k^* = \arg\min_k J^{(k)}$
  $\bar{U} \leftarrow \texttt{shift}(U^*, \tau)$ *// shift mean forward*
  $\Sigma \leftarrow \text{diag}\left(\text{Var}(\{U^{(k)}\}_{k=1}^{N_{\text{elite}}})\right)$
- Execute control $\boldsymbol{u}_\tau \leftarrow \texttt{get\_action}(U^*, \tau)$
- Update state history $\boldsymbol{h}_{\tau+1} \leftarrow \texttt{roll}(\boldsymbol{h}_\tau, \boldsymbol{x}_\tau)$

distribution of the SPC algorithm (e.g., $\pi_\phi(U)$ in CEM) with samples from $p_\theta(U \mid \boldsymbol{x}_\tau, \boldsymbol{h}_\tau)$. We call the first approach *GPC-Shoot* and the second approach *GPC-CEM*.

C. *GPC-CEM: Bootstrapping SPC with Flow-Matching*

Trained on a finite set of open-loop control sequences, the generative proposal distribution $p_\theta(U \mid \boldsymbol{x}_\tau, \boldsymbol{h}_\tau)$ inherits the generalization limitations of behavior cloning and model-based RL. This sensitivity to distributional shifts is something we also observe in our experiments with GPC-Shoot, where it manifests as degraded sample quality within regions underrepresented in the training data. In contrast, SPC adapts its sampling distribution online and becomes more robust in unseen situations, but remains myopic and computationally expensive. To balance these limitations, we bootstrap SPC with a flow-matching generative model that learns the dataset control distribution while preserving the adaptability of online sampling. We summarize our approach in Algorithm 1. At each control step, the algorithm begins by estimating the current state and shifting the current mean of the CEM sampling distribution forward in time. The key idea in GPC-CEM is to augment Gaussian CEM sampling with proposals from the generative model p_θ trained offline on control sequences that led to task success. The N_{elite} proposals with the lowest-cost rollouts are used to update the CEM distribution's $\pi_\phi(U)$ mean (the time-shifted lowest-cost proposal) and variance. Unlike standard CEM, the executed control is the best-performing candidate instead of the mean of the N_{elite} proposals. This allows GPC-CEM to better exploit multimodal proposals from the generative model rather than collapsing them to a single modality, efficiently guiding exploration while maintaining the adaptability of online optimization.

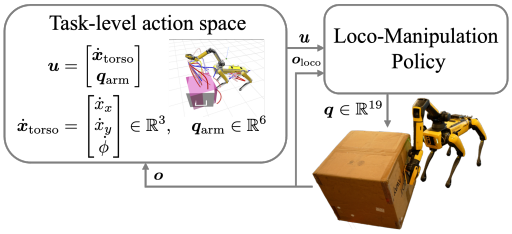


Fig. 2. The mapping from the high-level action space to the low-level spot loco-manipulation control space. The locomotion policy is pre-trained to follow the high-level commands while maintaining balance and stability. It is fixed throughout the full task planning and execution.

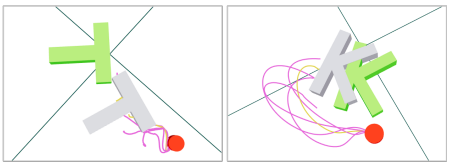


Fig. 3. **Push-T Task overview.** *Left:* Original Push-T task [9], where a circular robot is required to push a T-shaped block into a target pose, shown in green. *Right:* Modified task with a K- instead of T-block at the bottom.

D. *Application to Loco-Manipulation*

We apply our bootstrapped SPC framework to non-prehensile object pushing with a Spot quadruped robot to demonstrate versatile loco-manipulation skills. With 19 positional degrees of freedom (DoF), planning for this robot is computationally expensive and typically demands large sample sizes. To simplify our sampling process, we disentangle low-level locomotion control based on the work of [40] and sample only in the high-level task action space, as illustrated in Fig. 2. The high-level action space includes 9 DoF (3 for the torso, 6 for the arm)[1] and is mapped to the low-level commands by a pre-trained locomotion policy that ensures balance and stability while tracking high-level inputs. The low-level locomotion policy is fixed throughout task planning and execution. In addition to a lower-dimensional action space, this hierarchical control structure naturally provides more robustness to the low-level control and removes the need to introduce guidance terms that explicitly enforce smoothness and temporal consistency of the control sequences in the flow-matching process.

V. Experimental Results

We evaluate our proposed GPC framework across simulated and real-world continuous control tasks involving contact-rich, non-prehensile manipulation. Specifically, we benchmark performance on *i)* the well-known Push-T task with a 2-DoF circular robot, and *ii)* the loco-manipulation task introduced above. To guide our evaluation, we aim to answer the following key questions: *i)* How well does the learned generative model approximate the action proposal distribution captured by open-loop sampling-based MPC? *ii)* Does bootstrapping online MPC with a learned proposal

[1]We exclude the gripper DoF from the high-level action space for non-prehensile manipulation, but this and additional DoFs (e.g., torso height, pitch, roll) could be included without retraining the low-level policy.

Control frequency: 10 Hz *Time step* (Δt)*:* 0.01 s *Rollouts:* 32			
	Success rate (↑)	**Number of steps** (success only, (↓))	**CEM sample ratio**
Base Task: Push-T			
CEM Baseline	0.85 (0.83, 0.88)	1037.57 ± 526.40	–
MPPI Baseline	0.62 (0.59, 0.65)	1634.15 ± 492.42	–
Dial-MPC [18]	0.86 (0.83, 0.88)	1197.07 ± 461.16	–
GPC-Shoot (CVAE)	0.970 (0.951, 0.982)	985.31 ± 475.07	–
GPC-Shoot (2)	0.718 (0.702, 0.734)	1277.51 ± 572.80	–
GPC-Shoot (10)	0.992 (0.988, 0.995)	608.58 ± 291.48	–
GPC-CEM (2)	0.980 (0.980, 0.985)	932.40 ± 449.95	0.33 ± 0.11
GPC-CEM (10)	**0.998** (0.996, 0.999)	**591.11** ± 267.20	0.33 ± 0.10
Horizon Ablation: using 1 secs. instead of 3 secs. at inference time			
CEM Baseline	0.78 (0.76, 0.81)	1093.50 ± 523.09	–
MPPI Baseline	0.68 (0.65, 0.71)	1365.65 ± 463.48	–
Dial-MPC [18]	0.84 (0.81, 0.86)	945.76 ± 403.05	–
GPC-Shoot (CVAE)	0.71 (0.67, 0.75)	1209.23 ± 581.55	–
GPC-Shoot (10)	0.84 (0.83, 0.85)	978.96 ± 543.72	–
GPC-CEM (10)	**0.96** (0.95, 0.97)	**890.42** ± 466.94	0.42 ± 0.08
Task Variation: Push-K			
CEM Baseline	0.55 (0.52, 0.58)	1143.21 ± 565.90	–
MPPI Baseline	0.34 (0.31, 0.36)	1707.92 ± 501.53	–
Dial-MPC [18]	0.56 (0.52, 0.59)	1333.13 ± 484.97	–
GPC-Shoot (CVAE)	0.51 (0.46, 0.55)	1055.39 ± 539.90	–
GPC-Shoot (10)	0.89 (0.88, 0.90)	1015.15 ± 527.04	–
GPC-CEM (10)	**0.96** (0.95, 0.96)	**887.77** ± 482.02	0.41 ± 0.10

TABLE I

SIMULATION RESULTS FOR THE PUSH-T TASK, INCLUDING A HORIZON ABLATION AND A TASK VARIATION WITH A K- INSTEAD OF A T-BLOCK.

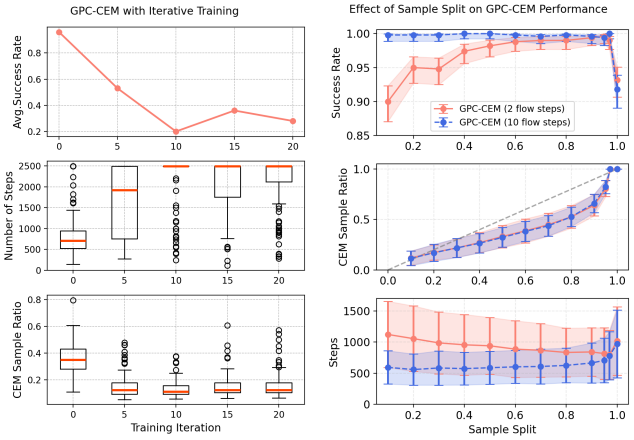


Fig. 4. **Push-T Simulation Studies.** *Left:* Evaluation of intermediate models from the iterative training procedure of [14]. *Right:* Ablation study on sample split.

distribution improve *task performance* and *generalization to task variations* under constrained computational budgets?

We use conditional flow matching to train a generative model over SPC control sequences with a Multi-Layer Perceptron (MLP) as the underlying architecture. We also baseline the flow model against a CVAE trained on the same data. We consider both direct sampling from the learned model (*GPC-Shoot*) and the bootstrapped version that combines it with CEM-based online planning (*GPC-CEM*). We compare our approach to standard CEM, model predictive path integral control (MPPI), [16] and DialMPC [18] — a more recent SPC approach building on MPPI. We only compare against DialMPC and GPC-Shoot with a CVAE on Push-T; the former's inner optimization loop makes it unsuitable for our real-time loco-manipulation examples, while the latter proved inferior to flow matching. All methods are implemented in Python using `judo` [41] as a unified interface for defining custom tasks and controllers. For each task, we evaluate all methods with the same number of rollouts per iteration, control frequency, and respective cost function. In addition, we report results for the GPC-methods across three different model seeds to account for the stochasticity during training. We set the CEM-sample ratio in GPC-CEM, i.e. N_{CEM}/N, to 0.5 for both tasks.

A. Push-T Task

This task requires a 2-DoF circular robot to push a T-shaped block to a specified goal pose. Due to its sparse rewards and multi-modality, it serves as a popular benchmark for evaluating generative control policies. We also evaluate the adaptability of GPC to unseen task variations by running it on a variant using a K-shaped block (Push-K) with different object dynamics. In this setting, we reuse the generative model trained on Push-T to bootstrap SPC for Push-K without retraining , showcasing GPC's ability to generalize across task variations. Table I summarizes the simulation results. We report success rates with Wilson 95% confidence intervals and average completion times (for successful runs) with respective standard deviations based on 1000 trials per method. Success is defined as achieving at least 90% coverage of the target pose within 2500 time steps (0.01 s each). For GPC-CEM, we additionally report the average CEM sample ratio and standard deviation, indicating how often samples were selected from the CEM distribution over the learned proposal distribution.

Both CEM and DIAL-MPC [18] achieve high success rates ($\geq 85\%$), demonstrating the strength of sampling-based methods. DIAL-MPC's gains over CEM are marginal and come at higher computational cost, so we use CEM for data collection. MPPI achieves 62% success, though better tuning may close this gap. Flow-based GPC-Shoot improves with more denoising steps (indicated in parentheses): 99% success with 10 steps vs. 71% with 2, while CVAE-based GPC-Shoot achieves 97% success. Similarly, GPC-CEM with 10 steps not only achieves 99.8% success but also reduces completion time by nearly 50% compared to 2 steps. Notably, GPC-CEM remains robust under reduced horizons (1s vs. 3s), maintaining 96% success versus 78% for CEM. This suggests that our method enables non-myopic planning under tighter computational constraints. In the Push-K generalization task, GPC-CEM again outperforms all baselines, achieving 96% success compared to 55% for CEM, indicating strong transferability of the learned proposal distribution. This is a notable insight, as both CVAE-based GPC-Shoot and CEM performance drops significantly without any changes to the cost function, highlighting that the flow-based learned distribution captures structural priors useful across tasks.

Comparison to Iterative Training Procedure [14]: We acknowledge the conceptual similarity between our approach and that of Kurtz et al. [14], who also combine generative

modeling with SPC. However, their method adopts an iterative training procedure that alternates between data collection and model updates, similar to expert iteration in reinforcement learning [42]. This setup is motivated by the assumption that SPC data is too noisy to directly train a generative model; hence, each data collection iteration bootstraps SPC with a partially trained flow-matching model to improve the subsequent training distribution. In our experiments on the Push-T task, however, this iterative procedure did not improve performance (see Fig. 4). In fact, success rates decline over training iterations. In a qualitative analysis, we find that the resulting policies tend to collapse to small, random movements that fail to complete the task. We interpret this as the iterative training procedure gradually diminishing the multi-modality of the learned proposal distribution. Consistent with this interpretation, we also observe a decreasing CEM sample ratio over training iterations, suggesting that the learned proposal distribution converges to mimic the CEM sampling distribution and reduces their complementarity. In contrast, our method trains a generative model directly on open-loop control sequences from SPC, without requiring iterative retraining. Despite the noisy data, our model achieves up to 99.2% success (GPC-Shoot with 10 denoising steps) and already reaches 96% when bootstrapped with CEM after a single training round.

Ablations: We conduct three ablation studies to further analyze the performance of GPC-CEM and GPC-Shoot on the simulated Push-T task. **Ablation 1** measures the effect of the sample split between samples drawn from the CEM distribution and the learned proposal distribution. We vary the sample split N_{CEM}/N from 0.1 to 1.0 (only CEM). As shown in Fig. 4, we find that increasing the number of CEM samples improves or maintains the success rate until exclusively using CEM samples, where performance drops significantly. This highlights that the learned proposal distribution contributes valuable samples that both complement and strengthen the CEM distribution. We further observe that the empirical CEM sample ratio grows sub-linearly with respect to the specified split, indicating that even a relatively small fraction of learned proposal samples disproportionately contribute to the overall sampling process. **Ablation 2** analyzes the impact of the dataset size used to train the flow-matching model. To assess how many demonstrations are needed to train an effective proposal model, we vary the dataset size from 100 to 1000 MPC rollouts. Performance improves rapidly with data and surpasses 90% success with only 200 rollouts, after which performance gains steadily saturate. This shows that the proposal model can be trained in a sample efficient manner requiring just a few hundred trajectories, whereas reinforcement learning methods typically need orders of magnitude more interaction to achieve comparable performance. **Ablation 3** assesses model architecture by comparing flow model performance against a conditional variational auto-encoder using GPC-Shoot. Table I shows the ten-step flow model outperformed the CVAE for each task and variation with a similar number of steps. Along with its degraded performance on the Push-K task, the CVAE's struggle to adjust to a different horizon length was characteristic of its documented difficulties with multi-modal generalization and domain adaptation [43]. Due to its higher performance and robust generalization, we selected flow matching as our generative model.

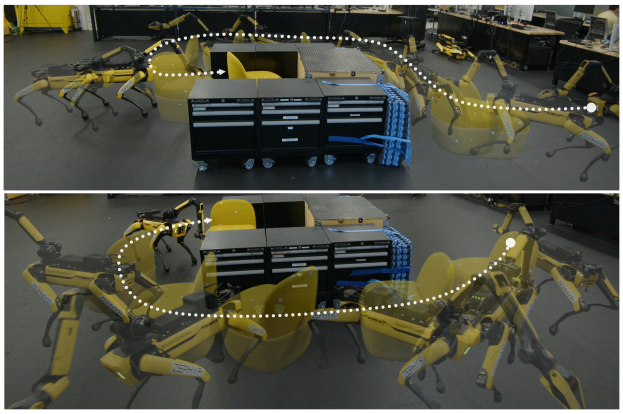


Fig. 5. **Spot Loco-Manipulation Experiment:** Qualitative examples of two successful Spot loco-manipulation task runs in the real world with GPC-CEM. Both images show a overlay of several snapshots of the trajectories of the robot and the chair.

B. Spot Loco-Manipulation

In this task, Spot must push a chair to a goal pose located behind a C-shaped obstacle (Fig. 5). The robot and chair are always initialized randomly on the opposite side of the obstacle, creating a local minimum that requires navigating around it to succeed. The task is further complicated by the high-dimensional action space and contact dynamics. Solving this with SPC requires long horizons and large sample sizes, both of which increase computational cost. A trial is considered successful if the chair's position error is below 0.15 m and its yaw is within 50 degrees of the target.

Simulation: We first evaluate the task in simulation to enable larger-scale testing. Results are summarized in Table II. Each baseline is run for 100 trials, and GPC methods are evaluated with 3 model seeds (100 trials each). We omit DIAL-MPC due to its inner-loop optimization being too slow for real-time use in this task. As in Push-T, we report success rate, average completion steps (for successful runs), and CEM sample ratio. GPC-CEM outperforms all baselines, achieving up to 83% success with fewer executed steps. In contrast, CEM alone reaches only 33%, often failing due to limited horizon and sample budget. MPPI performs slightly better but remains unreliable under real-time constraints. GPC-CEM remains robust under reduced planning horizons (3 vs. 4 seconds), maintaining 60% success while baseline performance degrades. This reinforces that learned proposals can enhance planning in resource-limited settings. Interestingly, fewer denoising steps (2 vs. 10) yield better performance in this task for both GPC-Shoot and GPC-CEM. We attribute this to reduced sample diversity at higher step counts, which impairs exploration in tasks with deceptive local minima. We also observe higher CEM sample ratios in this task compared to Push-T, indicating that the learned

Control frequency: 5 Hz	*Time step (Δt):* 0.02 s	*Rollouts:* 32	
	Succ. rate (↑)	**Number of steps** (success only, (↓))	**CEM sample ratio**
Base Task: Spot Loco-Manipulation			
CEM Baseline	0.33 (0.28, 0.39)	1452.1 ± 448.2	–
MPPI Baseline	0.57 (0.51, 0.62)	1096.3 ± 360.6	–
GPC-Shoot (2)	0.18 (0.14, 0.23)	1544.6 ± 495.2	–
GPC-Shoot (10)	0.22 (0.18, 0.27)	1454.2 ± 511.5	–
GPC-CEM (2)	**0.83** (0.78, 0.87)	1125.9 ± 430.6	0.69 ± 0.10
GPC-CEM (10)	0.79 (0.74, 0.83)	**1073.9 ± 367.4**	0.66 ± 0.11
Horizon Ablation: using 3 secs. instead of 4 secs. at inference time			
CEM Baseline	0.26 (0.21, 0.31)	1494.6 ± 491.9	–
MPPI Baseline	0.29 (0.24, 0.34)	1177.3 ± 465.3	–
GPC-Shoot (2)	0.22 (0.18, 0.27)	1489.3 ± 498.7	–
GPC-CEM (2)	**0.60** (0.54, 0.65)	1135.3 ± 487.9	0.71 ± 0.08
Task Variation: Spot Loco-Manipulation with Obstacle Avoidance			
CEM Baseline	0.27 (0.22, 0.32)	1692.3 ± 439.7	–
MPPI Baseline	0.30 (0.25, 0.35)	1634.2 ± 473.9	–
GPC-Shoot (2)	0.03 (0.02, 0.06)	1764.5 ± 353.2	–
GPC-CEM (2)	**0.56** (0.50, 0.62)	**1421.1 ± 481.8**	0.74 ± 0.08

TABLE II

SIMULATION RESULTS FOR THE SPOT LOCO-MANIPULATION TASK, INCLUDING HORIZON ABLATION AND TASK VARIATION WITH ADDITIONAL OBSTACLE AVOIDANCE COST AT RUNTIME.

model alone (GPC-Shoot) is less accurate. Instead, it is most effective when used to augment CEM, highlighting the value of integrating learned proposals into online optimization rather than relying on them directly. Finally, we evaluate a task variant with an added obstacle avoidance cost to prevent collisions with the C-shaped obstacle. This is omitted from the base task to avoid biasing the MPC methods, but is essential for real-world deployment. In this setting, GPC-CEM still leads with a 56% success rate, outperforming MPPI (30%) and CEM (27%).

Real-World: We evaluate GPC-CEM and CEM on hardware including the obstacle avoidance cost in both cases. We rely on a Motion Capture system to track the object state. GPC-CEM is run with 2 denoising steps for 20 trials, while CEM is limited to 10 trials due to frequent damaging failures to the robot (e.g. repeated collisions with the obstacle as it fails to navigate around). **GPC-CEM** achieves a **60% success rate (12/20)**, while **CEM** succeeds in only **10% of trials (1/10)**. Qualitative examples for GPC-CEM are shown in Fig. 5; all other runs are included in the supplementary video. CEM failures consistently result in the local minimum caused by the C-shaped obstacle, as it lacks the guidance from the generative model to sample motions that navigate around it. GPC-CEM only encounters this failure in 4 of 20 trials. The remaining failures stem from two causes: (1) pushing the chair beyond the workspace due to the lack of workspace constraints in the cost, and (2) discrepancies between simulated and real chair behavior, especially assumptions about friction and contact such as when the chair's wheels can roll.

Computation Time: We find that the policy rollout accounts for over 90% of the total compute time and becomes the primary computation bottleneck, limiting the overall control frequency to 5 Hz. This overhead is primarily due to collision handling and contact dynamics in the physics engine.

VI. LIMITATIONS AND FUTURE WORK

Our framework does not explicitly address sim-to-real transfer, leaving it vulnerable to discrepancies between simulated and real-world dynamics. However, since it relies on offline data collection, it can be trained on domain-randomized data to improve robustness to variations in dynamics, actuator behavior, and sensor noise [14]. In addition, learning proposal distributions from a combination of simulated and real-world data could further enhance transferability and performance during hardware deployment. In this work, all experiments consider fixed goals in a world frame. We plan to extend our work to variable goals by transforming our data into goal-centric representations. The current system also does not use GPU acceleration for simulation or proposal inference, but this could be addressed in future work to enable faster online planning and data collection (particularly with larger sample sizes). Finally, the proposed approach is limited to state-based policies but can be distilled to vision-based policies by learning from observations collected while executing the state-based policy in the real world or simulation. Future work can explore how to integrate vision-based action proposal distributions with fast vision-based dynamics models for online predictive control [13]. Although our offline data collection already captures long-horizon structure by using extended SPC horizons, future work could also incorporate learned infinite-horizon value functions. Such value estimates would provide an additional source of global, task-level guidance, while our generative priors would continue to shape and improve the search over control sequences during online optimization.

VII. CONCLUSION

We presented GPC-CEM, a generative predictive control (GPC) framework that bootstraps sampling-based MPC (SPC) with conditional flow-matching trained on open-loop SPC control sequences. Our approach demonstrates that meaningful proposal distributions can be learned directly from noisy SPC data, without expert supervision or iterative refinement. We evaluated our method in two challenging settings; a simulated pushing benchmark and a real-world quadruped loco-manipulation task; and showed that it significantly improves sample efficiency and robustness. GPC-CEM achieves high success rates, remains effective under reduced planning horizons, and generalizes to task variations which introduce out-of-distribution conditions. These results highlight the effectiveness of integrating learned generative models into online optimization loops for efficient and adaptable real-time robot control.

REFERENCES

[1] A. H. Li, P. Culbertson, V. Kurtz, and A. D. Ames, "Drop: Dexterous reorientation via online planning," *arXiv preprint arXiv:2409.14562*, 2024.

[2] T. Howell, N. Gileadi, S. Tunyasuvunakool, K. Zakka, T. Erez, and Y. Tassa, "Predictive sampling: Real-time behaviour synthesis with mujoco," *arXiv preprint arXiv:2212.00541*, 2022.

[3] J. Jankowski, L. Brudermüller, N. Hawes, and S. Calinon, "Vp-sto: Via-point-based stochastic trajectory optimization for reactive robot behavior," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 125–10 131.

[4] M. Bhardwaj, B. Sundaralingam, A. Mousavian, N. D. Ratliff, D. Fox, F. Ramos, and B. Boots, "Storm: An integrated framework for fast joint-space model-predictive control for reactive manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 750–759.

[5] B. Ichter, J. Harrison, and M. Pavone, "Learning sampling distributions for robot motion planning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7087–7094.

[6] A. Fishman, A. Murali, C. Eppner, B. Peele, B. Boots, and D. Fox, "Motion policy networks," in *conference on Robot Learning*. PMLR, 2023, pp. 967–977.

[7] G. Zhou, S. Swaminathan, R. V. Raju, J. S. Guntupalli, W. Lehrach, J. Ortiz, A. Dedieu, M. Lázaro-Gredilla, and K. Murphy, "Diffusion model predictive control," *arXiv preprint arXiv:2410.05364*, 2024.

[8] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π0: A vision-language-action flow model for general robot control, 2024," *arXiv preprint arXiv:2410.24164*, 2024.

[9] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, 2023.

[10] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

[11] N. Hansen, H. Su, and X. Wang, "Td-mpc2: Scalable, robust world models for continuous control," *arXiv preprint arXiv:2310.16828*, 2023.

[12] I. Dadiotis, M. Mittal, N. Tsagarakis, and M. Hutter, "Dynamic object goal pushing with mobile manipulators through model-free constrained reinforcement learning," *arXiv preprint arXiv:2502.01546*, 2025.

[13] H. Qi, H. Yin, Y. Du, and H. Yang, "Strengthening generative robot policies through predictive world modeling," *arXiv preprint arXiv:2502.00622*, 2025.

[14] V. Kurtz and J. W. Burdick, "Generative predictive control: Flow matching policies for dynamic and difficult-to-demonstrate tasks," *arXiv preprint arXiv:2502.13406*, 2025.

[15] Y. Wang, H. Guo, S. Wang, L. Qian, and X. Lan, "Bootstrapped model predictive control," *arXiv preprint arXiv:2503.18871*, 2025.

[16] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control: From theory to parallel computation," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 344–357, 2017.

[17] M. Kobilarov, "Cross-entropy motion planning," *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 855–871, 2012.

[18] H. Xue, C. Pan, Z. Yi, G. Qu, and G. Shi, "Full-order sampling-based mpc for torque-level locomotion control via diffusion-style annealing," *arXiv preprint arXiv:2409.15610*, 2024.

[19] C. Pan, Z. Yi, G. Shi, and G. Qu, "Model-based diffusion for trajectory optimization," *Advances in Neural Information Processing Systems*, vol. 37, pp. 57 914–57 943, 2024.

[20] T. Pang and R. Tedrake, "A convex quasistatic time-stepping scheme for rigid multibody systems with contact and friction," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6614–6620.

[21] A. K. Jain, V. Mohta, S. Kim, A. Bhardwaj, J. Ren, Y. Feng, S. Choudhury, and G. Swamy, "A smooth sea never made a skilled `sailor`: Robust imitation via learning to search," *arXiv preprint arXiv:2506.05294*, 2025.

[22] M. Dalal, J. Yang, R. Mendonca, Y. Khaky, R. Salakhutdinov, and D. Pathak, "Neural mp: A generalist neural motion planner," *arXiv preprint arXiv:2409.05864*, 2024.

[23] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters, "Motion planning diffusion: Learning and planning of robot motions with diffusion models," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 1916–1923.

[24] J. Urain, A. T. Le, A. Lambert, G. Chalvatzaki, B. Boots, and J. Peters, "Learning implicit priors for motion optimization," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 7672–7679.

[25] J. Carius, F. Farshidian, and M. Hutter, "Mpc-net: A first principles guided policy search," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2897–2904, 2020.

[26] T.-Y. Huang, A. Lederer, N. Hoischen, J. Brüdigam, X. Xiao, S. Sosnowski, and S. Hirche, "Toward near-globally optimal nonlinear model predictive control via diffusion models," *arXiv preprint arXiv:2412.08278*, 2024.

[27] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.

[28] J. Sacks and B. Boots, "Learning sampling distributions for model predictive control," in *Conference on Robot Learning*. PMLR, 2023, pp. 1733–1742.

[29] T. Power and D. Berenson, "Learning a generalizable trajectory sampling distribution for model predictive control," *IEEE Transactions on Robotics*, 2024.

[30] A. Jordana, S. Kleff, A. Haffemayer, J. Ortiz-Haro, J. Carpentier, N. Mansard, and L. Righetti, "Infinite-horizon value function approximation for model predictive control," *IEEE Robotics and Automation Letters*, 2025.

[31] D. Hoeller, F. Farshidian, and M. Hutter, "Deep value model predictive control," in *Conference on robot learning*. PMLR, 2020, pp. 990–1004.

[32] K. Lowrey, A. Rajeswaran, S. Kakade, E. Todorov, and I. Mordatch, "Plan online, learn offline: Efficient learning and exploration via model-based control," *arXiv preprint arXiv:1811.01848*, 2018.

[33] N. Hatch and B. Boots, "The value of planning for infinite-horizon model predictive control," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7372–7378.

[34] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.

[35] R. Y. Rubinstein and D. P. Kroese, *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2004.

[36] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," *arXiv preprint arXiv:2210.02747*, 2022.

[37] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[38] A. Tong, K. Fatras, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, G. Wolf, and Y. Bengio, "Improving and generalizing flow-based generative models with minibatch optimal transport," *arXiv preprint arXiv:2302.00482*, 2023.

[39] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 2786–2793.

[40] X. Zhu, Y. Chen, L. Sun, F. Niroui, S. Le Cleac'h, J. Wang, and K. Fang, "Versatile loco-manipulation through flexible interlimb coordination," *arXiv preprint arXiv:2506.07876*, 2025.

[41] A. Li, S. L. Cleac'h, B. Hung, A. D. Ames, J. Wang, and P. Culbertson, "Judo: A user-friendly open-source package for sampling-based model predictive control," in *Proceedings of the Workshop on Fast Motion Planning and Control in the Era of Parallelism at Robotics: Science and Systems (RSS)*, 2025. [Online]. Available: https://github.com/bdaiinstitute/judo

[42] T. Anthony, Z. Tian, and D. Barber, "Thinking fast and slow with deep learning and tree search," *Advances in neural information processing systems*, vol. 30, 2017.

[43] I. Daunhawer, T. M. Sutter, K. Chin-Cheong, E. Palumbo, and J. E. Vogt, "On the limitations of multimodal VAEs," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=w-CPUXXrAj

4.1 Supplementary Discussion

Our Generative Predictive Control (GPC) framework sits at the intersection of reinforcement learning (RL), imitation learning, and sampling-based model predictive control (MPC). From an RL perspective, the approach is most closely related to *KL-regularised RL* approaches, which incorporate expert policies as priors to stabilize training and improve sample efficiency, e.g., TRPO (Schulman et al., 2015) and MPO (Abdolmaleki et al., 2018). These methods constrain a learned policy to remain close to a prior, i.e., expert policy, usually derived from demonstrations or previous rollouts. However, recent theoretical analysis by GX-Chen et al. (2025) shows that standard KL-regularised RL objectives lead to solutions that collapse to a single mode, even when the reward landscape itself is highly multimodal. This property significantly limits their usefulness in domains such as contact-rich manipulation and loco-manipulation, where multiple qualitatively distinct solution strategies naturally emerge and are essential for robustness. The GPC framework also differs from *residual RL*, where a learned residual policy is added on top of a classical controller to compensate for modelling errors or unmodelled dynamics (Johannink et al., 2019). While residual RL benefits from incorporating a stabilizing base controller, it ultimately learns a standalone reactive policy and therefore inherits the challenges faced by RL in long-horizon, discontinuous tasks, including sparse exploration, brittle generalisation, and the difficulty of representing diverse action sequences.

In this context, our approach provides a novel alternative that combines the strengths of both KL-regularised RL and imitation learning from an MPC expert. Similar to KL-regularised RL, GPC leverages a learned prior over successful behaviors. And like imitation learning, this prior is trained on expert data; in our case, trajectories generated by an MPC planner. Crucially, however, GPC does not attempt to replace the planner with a standalone policy. Instead, it learns a generative proposal distribution over full control sequences that bootstraps sampling-based MPC, while leaving the task objective, constraints, and online adaptability

entirely to the optimizer. This design preserves the inherent multimodality of the solution set, allowing MPC to choose among diverse behavioural modes at runtime. It also avoids distribution-shift issues that commonly arise when substituting the planner with an end-to-end RL or imitation-learned policy.

4.2 Limitations and Future Work

Robustness to Uncertainty A current limitation of our framework is that it does not explicitly account for uncertainties. However, the offline data collection stage naturally provides an opportunity for more principled uncertainty handling. In particular, candidate trajectories can be simulated under a range of environment parameterisations via domain randomisation (Tobin et al., 2017) and evaluated using different risk objectives, as outlined in Section 2.6.2. Learning from such data would allow the proposal distributions to implicitly capture the effects of uncertainty, without requiring explicit uncertainty models during online execution. Exploring such approaches represents an interesting direction for future work, and has recently been investigated in simulation by Kurtz and Burdick (2025).

On the Choice of Action Representation Another important design choice concerns the action representation. Recent work has shown that policy learning can benefit from equivariance through relative action spaces (Wang et al., 2025a; Chi et al., 2024; Jankowski et al., 2025b), where actions such as end-effector poses are expressed relative to the current pose. In our experiments, transforming both actions and observations into the robot frame did not yield performance improvements over absolute actions represented in world frame. We suspect this may be because our framework operates in a state-based setting, where the benefits of relative representations are less pronounced than in vision-based approaches. Future work should therefore explore potential benefits of relative action spaces in other tasks and embodiments to determine whether this limitation is specific to our setting or a more general characteristic of state-based policies.

Variable Horizon Lengths We have so far only considered fixed horizon lengths during both offline data collection and online execution. However, we believe that allowing for variable horizon lengths could further enhance the reactivity of the framework. Not all tasks require long-horizon planning, and in many scenarios, shorter horizons may suffice, which would reduce computation time and improve reactivity. In addition, especially tasks that have a natural termination point, such as reaching or grasping, could benefit from variable horizon lengths to avoid overshooting the goal. Prior work has explored this idea by treating the horizon length as a decision variable in trajectory optimisation (Shekhar, 2012). Similarly, recent work investigated learning a horizon length predictor for variable length diffusion planning (Liu et al., 2025). An alternative and potentially more natural approach lies in learning VP-STO trajectory representations, where horizon length is inherently flexible: via-points map to timing-optimal control sequences through the kino-dynamic constraints of the system, enabling variable horizon lengths by design. Consequently, the network architecture itself remains unaffected by changes in horizon length, since the number of via-points is fixed. In this work, however, we chose not to exploit VP-STO in order to keep the online optimisation as simple as possible and avoid the additional complexity of handling basis functions and latent-to-control mappings. Moreover, we aimed to study the effect of learned proposals in a more general sampling-based MPC setting, independent of the additional structure imposed by VP-STO.

Warm-Starting Flow-Models A common challenge in deploying generative models for real-time robot control is ensuring *temporal consistency* of the generated samples across inference steps (Zhao et al., 2023). Given the stochastic nature of generative models, not all samples are equally good at capturing the critical temporal dependencies (Torne et al., 2025). Similar to the concept of warm-starting in MPC, we can also warm-start the generative model at each MPC step with the best sample from the previous MPC step. Instead of initializing the flow generation process by sampling from a general source distribution, the goal is to find

a theoretically grounded way to initialise the sampling process with a sample $\boldsymbol{x}_0$ that is close to the best sample from the previous MPC step. In other words, this corresponds to moving the noise distribution closer to the prior distribution (Scholz and Turner, 2025). Beyond the scope of this thesis, we have briefly explored an approach to warm-start flow models based on Bayesian posterior inference, which we outline in the following. The warm-starting approach combines the learned flow velocity field $v_\theta(\boldsymbol{x}_t, t)$ with a structured prior velocity field $\boldsymbol{v}_{\text{prior}}(\boldsymbol{x}_t, t)$, derived from a Gaussian warm-start distribution $\mathcal{N}(\boldsymbol{\mu}_{\text{prior}}, \boldsymbol{\Sigma}_{\text{prior}})$. This prior flow is calculated by adopting the Conditional Flow Matching (CFM) framework (Lipman et al., 2024), assuming a linear probability path $\boldsymbol{x}_t = t\,\boldsymbol{x}_1 + (1-t)\,\boldsymbol{x}_0$, where the target $\boldsymbol{x}_1 \sim \mathcal{N}(\boldsymbol{\mu}_{\text{prior}}, \boldsymbol{\Sigma}_{\text{prior}})$ and the source $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. The theoretical conditional velocity field (cf Eq 2.6 in (Lipman et al., 2024)) is

$$\boldsymbol{u}_t(\boldsymbol{x}|\boldsymbol{x}_1) = \frac{\boldsymbol{x}_1 - \boldsymbol{x}}{1-t}. \tag{4.1}$$

The required prior velocity field $\boldsymbol{v}^{\text{prior}}$ is the expectation of this conditional velocity over the posterior $p(\boldsymbol{x}_1|\boldsymbol{x}_t)$, yielding:

$$\boldsymbol{v}_{\text{prior}}(\boldsymbol{x}_t, t) = \frac{\mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_t] - \boldsymbol{x}_t}{1-t} \tag{4.2}$$

The conditional mean, $\boldsymbol{\mu}_{1|t} = \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_t]$, is calculated using the properties of jointly Gaussian distributions, resulting in a standard Bayesian update form:

$$\boldsymbol{\mu}_{1|t} = \left(\boldsymbol{\Sigma}^{-1} + \frac{t^2}{(1-t)^2}\boldsymbol{I}\right)^{-1}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \frac{t}{(1-t)^2}\boldsymbol{x}_t\right) \tag{4.3}$$

Finally, the full warm-start velocity field $\boldsymbol{v}_{\text{post}}$ used for the Euler ODE step could be a heuristic linear combination of the learned and prior fields:

$$\boldsymbol{v}_{\text{post}}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{2}}\left(\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) + \boldsymbol{v}_{\text{prior}}(\boldsymbol{x}_t, t)\right) \tag{4.4}$$

We see this warm-start strategy as a promising direction to improve temporal consistency, as well as sample efficiency, by biasing the sampling process towards previously successful trajectories. While we have not defined a specific way to set the warm-start distribution parameters $\boldsymbol{\mu}_{\text{prior}}$ and $\boldsymbol{\Sigma}_{\text{prior}}$, a natural choice would

be to set $\boldsymbol{\mu}_{\text{prior}}$ to the best time-shifted sample from the previous MPC step and $\boldsymbol{\Sigma}$ to a scaled identity matrix, where the scaling factor controls the exploration-exploitation trade-off. However, we have not empirically validated this method, and the theoretical properties of combining velocity fields in this manner remain an open question. In particular, the linear combination does not preserve the optimal transport structure of the individual velocity fields, and the resulting probability path may not correspond to a well-defined interpolation between the prior and learned distributions. Future work should investigate theoretically grounded approaches to velocity field composition. In addition, while warm-starting could be beneficial for temporal consistency and faster convergence, it may also reduce the diversity of samples and thus the exploration capabilities of the generative model. A heuristic approach could be to only warm-start a fraction of the samples at each MPC step, while sampling the rest from the original source distribution. Future work should investigate this trade-off between exploration and exploitation in more detail. We also note that this is conceptually similar to guidance in diffusion models (Dhariwal and Nichol, 2021), which has been explored by Yamada et al. (2025b) in the context of diffusion-based trajectory optimisation for deformable object manipulation. Recent work by Feng et al. (2025) has also explored guidance in the context of flow models, but focuses on guiding the sampling process towards specific target states, rather than warm-starting from a previous sample.

Cost Tuning Finally, we note that the proposed framework still requires task-specific cost function tuning, a long-standing challenge in MPC. A promising direction for future work is to combine our framework with recent advances in cost function learning. This can be achieved by learning cost-shaping terms (Tamar et al., 2017; Yamada et al., 2025a), terminal costs (Baltussen et al., 2025; Zhong et al., 2013), or even full cost approximations (Hansen et al., 2022; Finn et al., 2016), using either expert demonstrations or experience data from interaction, e.g., through a replay buffer from an RL agent. However, learning cost functions for MPC remains difficult, as the optimiser may sample out-of-distribution actions,

leading to unreliable estimates and degraded performance (Jawale et al., 2025). While value function learning has been extensively studied in approximate dynamic programming and reinforcement learning with closed-loop data (Bertsekas, 2019), its integration with MPC is less developed and presents an interesting research opportunity. For further discussion on MPC with value function approximations and its connections to RL, we refer the reader to Banker et al. (2025) and Mesbah et al. (2022). In the context of our proposed generative predictive control framework, this entails interesting research questions on how to best integrate cost function learning with data generation, as well as in the MPC loop itself. Given that we learn from open-loop control sequences generated by the MPC itself, instead of from expert demonstrations, i.e., closed-loop trajectories, replacing the cost function with a learned value function approximation is less straightforward.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

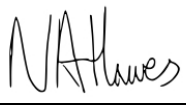
Title of Paper	Generative models from and for sampling-based MPC: A bootstrapped approach for adaptive contact-rich manipulation	
Publication Status	□ Published	□ Accepted for Publication
	☑ Submitted for Publication	□ Unpublished and unsubmitted work written in a manuscript style
Publication Details	Brudermüller, L., Hung, B., Zhu, X., Wang, J., Hawes, N., Culbertson, P., and Le Cleac'h, S. (2025). Generative models from and for sampling-based MPC: A bootstrapped approach for adaptive contact-rich manipulation. Manuscript under review at IEEE Robotics and Automation	

Student Confirmation

Student Name:	Lara Brudermüller		
Contribution to the Paper	– The paper was developed as part of an internship at the Robotics and AI institute in Boston. – I led the development of the idea in collaboration with Preston Culbertson and Simon Le Cleac'h. – I implemented the algorithm, as well as the training pipelines and benchmarking code. – I designed all experiments. Brandon Hung then helped putting them on hardware. – I led the writing and got feedback from Nick Hawes, Simon Le Cleac'h, Preston Culbertson and Brandon Hung.		
Signature		Date	September 17, 2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Professor Nick Hawes	
Supervisor comments I confirm that Lara made a substantial contribution to this publication, and that the description above is accurate.		
Signature		Date: September 17, 2025

This completed form should be included in the thesis, at the end of the relevant chapter.

5

Bridging Reactive and Robust Control through Chance-Constrained MPC

Publication Note

This chapter presents the work from the following submission:

Brudermüller, L., Berger, G. O., Jankowski, J., Bhattacharyya, R., Calinon, S., Jungers, R. M., and Hawes, N. (2025a). CC-VPSTO: Chance-Constrained Via-Point-Based Stochastic Trajectory Optimisation for Online Robot Motion Planning under Uncertainty. Manuscript under review at *The International Journal of Robotics Research*

Supplementary Material

Webpage with supplementary videos related to this chapter is available at:
`https://sites.google.com/oxfordrobotics.institute/cc-vpsto`.

In the previous two chapters, we focused on implicit uncertainty handling through reactive control with sampling-based MPC methods that can handle the discontinuities and non-smoothness inherent to contact-rich manipulation tasks. Given the assumption of accurate state estimation and perfect knowledge of the system dynamics, the primary goal of these works was to enhance the reactivity of the controller, enabling it to adapt effectively to disturbances and modelling errors through implicit uncertainty handling, rather than relying on explicit probabilistic reasoning or uncertainty propagation. Reactivity entails the ability to solve the underlying optimal control problem quickly and reliably at each time step, thereby

adapting control actions based on the latest state information. In Chapter 3, we introduced a low-dimensional trajectory representation that maps to continuous, timing-optimal trajectories and allows the incorporation of informative Gaussian priors, thereby improving the quality of the solutions found within the limited time budget available for online replanning. In Chapter 4, we further enhanced the reactivity of sampling-based MPC methods by learning proposal distributions from offline data to guide the MPC sampling process toward promising regions of the solution space, improving the efficiency of the sampling process and enabling the controller to find high-quality solutions more quickly.

However, even if the controller is sufficiently reactive to handle any disturbance, modelling error or sudden change in the environment, the resulting behaviour will most likely not be optimal with respect to the original task objective in the presence of uncertainty. For instance, in contact-rich manipulation tasks such as pushing or grasping, the stochastic nature of contact dynamics, arising from variations in friction, surface compliance, or unmodelled stick–slip transitions, can lead a purely reactive controller to generate suboptimal or inconsistent motions. While such a controller may recover from unexpected contact outcomes, it cannot proactively plan actions that account for the distribution of possible contact interactions, and thus may exhibit inefficient or unstable behaviour. In contrast, a controller that explicitly reasons about uncertainty can plan actions that balance task performance with robustness to uncertain outcomes, leading to more reliable and efficient behaviour.

In this chapter, we therefore focus on enhancing the *robustness* of sampling-based MPC methods to uncertainty by *explicitly* reasoning about uncertainty in the planning process. This step complements the previous chapters by addressing a key limitation of purely reactive approaches and contributes to the overarching goal of this thesis: developing planning and control methods that bridge the gap between fast, adaptive control and principled decision-making under uncertainty. Consequently, we now consider *stochastic* system and environment dynamics (cf. Eq. 2.18), in contrast to the *deterministic* dynamics assumed in Chapters 3 and 4.

Uncertainty-aware planning and control is a well-studied problem in the literature, with approaches ranging from robust MPC (RMPC) (Ben-Tal and Nemirovski, 1998) to stochastic MPC (SMPC) (Mesbah, 2016) and belief-space planning (Platt et al., 2010), as discussed in Section 2.6. We have noted that RMPC methods, which plan for the worst-case scenario, can be overly conservative and lead to suboptimal performance in practice. In contrast, SMPC methods, which incorporate *chance constraints* to probabilistically bound constraint violations, offer a more balanced approach that can achieve better performance while still ensuring safety with high probability. Yet, both categories of methods typically require certain assumptions about the system dynamics and uncertainty distributions in order to be computationally tractable, thereby limiting their applicability. This work aims to develop a more general approach that minimises such assumptions, yielding a task-agnostic framework applicable to a wide range of robotic systems and tasks without task-specific tuning or prior knowledge about the uncertainty characteristics. All we require is *sampling-based access* to the probabilistic uncertainty distributions, i.e., the ability to draw samples from the stochastic system, without assuming a specific parametric form or known moments of the distributions. We summarise this setting in Table 1.2, which is repeated below.

	Chapter 3	**Chapter 4**	**Chapter 5**	**Chapter 6**
Theme	Reactivity	Learning for Reactivity	Robustness	Exploration
Uncertainty Handling	Implicit	Implicit	Explicit	Explicit
Prior Knowledge of Uncertainty	None	None	Sampling-based access	Sampling-based access
Environment	Deterministic	Deterministic	Stochastic	Stochastic
Method	MPC	MPC	MPC	Belief-space control
Control Problem	Min. cost	Min. cost	Min. cost s.t. chance constraints	Max. information gain

Table 1.2: Theme, settings and methods for each chapter.

To this end, we extend VP-STO from Chapter 3 to handle chance constraints, resulting in the *Chance-Constrained VP-STO (CC-VPSTO)* algorithm. CC-VPSTO combines the benefits of the low-dimensional trajectory representation and informative priors for reactive robot behaviour from VP-STO with explicit uncertainty handling through chance constraints, enabling the planner to generate robust motions that satisfy safety constraints with high probability. Without loss of generality, we demonstrate the proposed approach in obstacle-avoidance experiments. While the broader focus of this thesis lies in contact-rich manipulation, we chose obstacle avoidance as a representative task to evaluate the proposed method, as the environment dynamics in obstacle interactions can be highly stochastic and even multi-modal, particularly when involving dynamic agents such as humans. This makes obstacle avoidance an informative and challenging setting for studying explicit uncertainty handling, whereas contact dynamics, though uncertain, tend to be less variable in comparison.

Finally, we note that the Appendix of the paper manuscript is included at the end of this thesis as Appendix B.

Journal Title
XX(X):1–22

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

CC-VPSTO: Chance-Constrained Via-Point-Based Stochastic Trajectory Optimisation for Online Robot Motion Planning under Uncertainty

Lara Brudermüller[1], Guillaume O. Berger[2], Julius Jankowski[3], Raunak Bhattacharyya[1], Sylvain Calinon[3], Raphaël M. Jungers[2], and Nick Hawes[1]

Abstract

Reliable robot autonomy hinges on decision-making systems that account for uncertainty without imposing overly conservative restrictions on the robot's action space. We introduce *Chance-Constrained Via-Point-Based Stochastic Trajectory Optimisation (CC-VPSTO)*, a real-time capable framework for generating task-efficient robot trajectories that satisfy constraints with high probability by formulating stochastic control as a chance-constrained optimisation problem. Since such problems are generally intractable, we propose a deterministic surrogate formulation based on Monte Carlo sampling, solved efficiently with gradient-free optimisation. To address bias in naïve sampling approaches, we quantify approximation error and introduce padding strategies to improve reliability. We focus on three challenges: (i) sample-efficient constraint approximation, (ii) conditions for surrogate solution validity, and (iii) online optimisation. Integrated into a receding-horizon MPC framework, CC-VPSTO enables reactive, task-efficient control under uncertainty, balancing constraint satisfaction and performance in a principled manner. The strengths of our approach lie in its generality, *i.e.*, no assumptions on the underlying uncertainty distribution, system dynamics, cost function, or the form of inequality constraints; and its applicability to online robot motion planning. We demonstrate the validity and efficiency of our approach in both simulation and on a Franka Emika robot. Videos and additional material are made available here.

Keywords

Chance-Constrained Optimisation, Stochastic Model Predictive Control, Trajectory Optimisation

1 Introduction

Uncertainty is inherent to most real-world robotics applications, arising from noisy sensors, imprecise actuators, and incomplete or evolving knowledge of the environment. Effectively managing this uncertainty is essential for achieving reliable and efficient robot behaviour, particularly in online motion planning tasks that require fast adaptation to new information. In this work, we adopt a *chance-constrained* perspective, where constraints such as collision avoidance (cf. Fig. 1), force limits, or task completion cannot be guaranteed but must instead be satisfied with high probability (Prékopa 2013; Dai et al. 2019). Unlike traditional robust control methods (Köhler et al. 2023; Majumdar and Tedrake 2017; Badings et al. 2023) that optimise for the worst-case scenario under *bounded uncertainty*, chance constraints enable a more general, *probabilistic* treatment of uncertainty (Margellos et al. 2014; Schildbach et al. 2014), allowing for more explicit trade-offs between constraint satisfaction and task efficiency.

Crucially, we are interested in an *online* robot motion planning setting where constraint violations are undesirable but not catastrophic, and where performance (*e.g.*, motion duration) remains important. Our objective is to trade off constraint satisfaction and task performance in a principled manner that avoids unnecessary conservatism. While Model Predictive Control (MPC) can implicitly provide some robustness via frequent replanning, it typically relies on deterministic models, leading to brittle, myopic behaviour in stochastic settings. Incorporating probabilistic information directly into the control loop remains a key challenge, but is essential for enabling more flexible and robust decision-making in uncertain environments.

Chance-constrained formulations, which require that constraints must be satisfied with high probability (*e.g.*, at least 95%), offer a natural solution but are in general intractable (Blackmore et al. 2010). One common strategy is to approximate the chance constraint and reformulate the problem as a deterministic surrogate that can be addressed using standard optimisation techniques. However, identifying a suitable approximation is often non-trivial. Common approaches either introduce significant conservatism at the cost of task efficiency (Lew et al. 2023; Calafiore and Campi 2006), or, like naïve sample

[1]Oxford Robotics Institute, University of Oxford, UK
[2]ICTEAM, UCLouvain, Belgium
[3]Idiap Research Institute & Ecole Polytechnique Fédérale de Lausanne (EPFL), CH

Corresponding author:
Lara Brudermüller, Oxford Robotics Institute,
23 Banbury Rd, Oxford OX2 6NN, UK
Email: larab@robots.ox.ac.uk

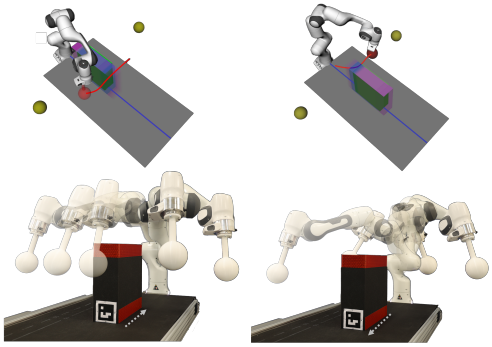


Figure 1. ***Robot experiment.*** The robot must move its ball-shaped end effector from a start to a goal point (yellow spheres) across a conveyor belt, while avoiding a box obstacle moving stochastically on the belt. Depending on the anticipated box motion, the robot can pass in front (left) or behind (right). This task requires online, reactive motion generation that balances constraint satisfaction with task efficiency, aiming to reach the goal as quickly as possible under uncertainty.

average approximation (SAA) methods (Shapiro et al. 2021; Shapiro 2003; Pagnoncelli et al. 2009), are overly optimistic under limited samples, biasing solutions toward regions that appear feasible in the surrogate problem but violate the true constraint (Homem-de Mello and Bayraksan 2014), a phenomenon we also observed in our experiments (cf., *e.g.*, Fig. 10).

Towards this end, we propose **CC-VPSTO** (Chance-Constrained Via-Point-Based Stochastic Trajectory Optimization), a novel Monte Carlo-based framework for *online* robot motion planning *under uncertainty*. CC-VPSTO systematically balances the inherent trade-offs between probabilistic constraint satisfaction, solution quality, and computational efficiency by focusing on three key challenges: *a)* given a number of samples and a *confidence level*, selecting a statistical *padding*, *i.e.*, a margin on the empirical constraint, that ensures the sample-based solution satisfies the true chance constraint with high confidence; *b)* keeping the padding small to avoid excessive conservatism; and *c)* solving the problem efficiently in real time with few samples. Let us emphasize that *a)*, *b)*, and *c)* are competing objectives since few samples usually result in overly conservative padding (such as in scenario optimisation approaches; *e.g.*, Calafiore and Campi 2006) or overly optimistic solutions (such as in naïve sample average approximation approaches; *e.g.*, Shapiro et al. 2021). In summary, the main contributions of this work are:

1. A new surrogate formulation for chance-constrained trajectory optimisation, designed specifically for online motion planning. It maintains low added conservatism even with small sample sizes, while accounting for approximation errors to avoid overly optimistic solutions.

2. A theoretical analysis of the surrogate's correctness. Under the assumption of independence between the solution and the samples, we show that the solution of the surrogate problem satisfies the original (intractable) chance constraint with high confidence. We further provide insights into why the approach remains effective in receding-horizon settings, such as MPC, where the independence assumption may not strictly hold.

3. The empirical evaluation of our approach across a range of challenging tasks. Even when used heuristically (i.e., without guaranteed independence), the surrogate performs reliably in both simulation and real-world robot experiments.

4. The integration of the surrogate formulation into *VP-STO*, an MPC framework for *online* reactive robot control (Jankowski et al. 2023), extending it to a stochastic setting. This enables receding-horizon control that effectively balances constraint satisfaction and task performance under uncertainty.

The key advantages of our approach are: *i)* flexibility to handle arbitrary uncertainty distributions, *ii)* compatibility with real-time MPC via parallelisable sampling and optimisation, and *iii)* support for general inequality constraints, such as collision avoidance, force limits, or performance objectives.

2 Related Work

Risk-averse planning and control methods in robotics aim to enforce constraints in the presence of uncertainty. These methods typically enforce these constraints by formulating them as chance constraints (CCs) or by using risk measures, such as Conditional Value-at-Risk (CVaR). When employed in a *online* receding horizon control scheme, these methods fall into the category of Stochastic Model Predictive Control (SMPC) (Heirung et al. 2018; Mesbah 2016). SMPC addresses optimal control problems for dynamical systems under stochastic uncertainty, while enforcing constraints that must be satisfied with high probability (*i.e.*, chance constraints). In the context of chance-constrained control, the literature distinguishes between two types of formulations for limiting constraint violations. These can be defined either *point-wise*, *i.e.*, imposing a separate probability bound at each time step, or *jointly*, where the constraint must hold over the entire (finite) time horizon with high probability. We focus on joint chance constraints in this work, as they are preferable in robotics where it is important to account for the cumulative effect of uncertainty over time. As discussed by Janson et al. (2017), many approaches bound the joint probability using either an *additive* approach, summing over all point-wise probabilities via Boole's inequality (*e.g.*, Ono and Williams (2008); Priore and Oishi (2023); Castillo-Lopez et al. (2020)), or a *multiplicative* approach, which involves explicitly constraining the product of the complements of point-wise probabilities (*e.g.*, Sun et al. (2016); Van Den Berg et al. (2011)). Yet, both of these strategies do not account for the time correlations of uncertainty, and thus may lead to over- or underestimation of the joint probability of constraint violation. We thus adopt the time-wise supremum approach from Lew et al. (2023), which evaluates only the maximum constraint violation over the entire time horizon for a given trajectory, thereby capturing time correlations effectively.

The main challenge in SMPC is evaluating the probability of constraint violation over the planning horizon. This

requires computing an expectation integral over time and space, which is typically intractable for general uncertainty distributions and constraint structures (Peña-Ordieres et al. 2020). As a result, SMPC must address two key questions: *i)* how to approximate or bound the probability of constraint violation in a tractable way, and *ii)* how to solve the resulting optimisation problem with minimal computational overhead for online control. Previous approaches to these questions proposed semidefinite programming formulations (Jasour et al. 2015) or constraint tightening (Alcan and Kyrki 2022; Ono and Williams 2008; Parsi et al. 2022). While effective in providing probabilistic guarantees on the satisfaction of chance constraints, they are typically tailored to very specific types of constraints, uncertainty distributions and/or system dynamics, thereby limiting their applicability to real robotics problems. As noted in Lew et al. (2023), there is still a lack of formulations and solution algorithms that are capable of capturing different sources of uncertainty as well as different types of constraints in a unified framework.

In contrast, sample-based methods offer a more general approach for approximating chance constraints, as they do not require any assumptions on the underlying probability distributions, as long as the number of samples is sufficiently large. In the sample-based setting, we can distinguish between *scenario optimisation* (Schildbach et al. 2014; de Groot et al. 2023) and *Monte Carlo* methods (Blackmore et al. 2010; Schmerling and Pavone 2016; Blackmore 2006). Both approaches use samples (aka. scenarios) to capture the underlying uncertainty. Scenario optimisation synthesises controls satisfying the constraint for each of the samples and relies on a well-established theory to identify the right sample size for a given confidence level (Calafiore and Campi 2006). However, these theoretical bounds are mostly limited to convex or quasi-convex problems (Calafiore 2010; Berger et al. 2021) and solutions are typically overly conservative, *i.e.*, they require much larger sample sizes than identified by empirical tests (Schildbach et al. 2014). Monte Carlo methods typically approximate the probability of constraint violation from the samples, rooted in the *sample average approximation (SAA)* approach (Shapiro et al. 2021; Shapiro 2003; Pagnoncelli et al. 2009). They are generally less conservative and can be used with arbitrary constraints and uncertainty models. However, without further adjustments, they do not provide finite-sample guarantees, but only asymptotic guarantees (Blackmore 2006), implying the requirement of large sample sets, and higher computational resources. The need for large sample sets is reinforced when the desired probability of constraint violation is low, as is commonly targeted in robotics applications. A remedy to this can be importance sampling (Schmerling and Pavone 2016), or data reduction methods based on parameter estimation of sample statistics, *e.g.*, through computing moments of the probability distribution of the uncertainty (Wang et al. 2020; Priore and Oishi 2023; Blackmore et al. 2006; Yan et al. 2018). Yet, the propagation of moments can be complex, and requires restrictive assumptions, such as Gaussianity. Alternatively, for collision avoidance, Trevisan et al. (2025) propose a naïve Monte Carlo approach that approximates the chance constraint using a fixed number of samples, increasing sample density by constraining the collision region across time steps. However, this method is tailored specifically to collision avoidance and does not account for the approximation error introduced by the finite sample size.

Other approaches have used SAA to approximate constraints on risk metrics like conditional Value-at-Risk (CVaR) instead (Lew et al. 2023; Yin et al. 2023; Nemirovski and Shapiro 2007). CVaR constraints are more conservative than chance constraints, as they account for tail events, but the resulting reformulation is smooth and convex, which enables the use of off-the-shelf optimisation tools, such as sequential convex programming (SCP). For instance, Lew et al. (2023) provide a general framework for risk-averse SMPC based on the combination of the SAA of CVaR constraints and concentration inequalities to bound the approximation error. However, their approach relies on strong continuity assumptions on the objective function and constraints, which may not hold in practice. Yin et al. (2023) use the SAA of CVaR constraints within a Model Predictive Path Integral (MPPI) controller but do not account for errors in the approximation of the CVaR constraint and limit the source of uncertainty to process noise. Finally, the work of Peña-Ordieres et al. (2020) reformulates the chance constraint as a quantile function and uses SAA to approximate it, which results in a formulation that is amenable to gradient-based optimisation methods. However, similar to Lew et al. (2023), their approach relies on continuity and differentiability assumptions of the constraint function.

Our main observation across chance constrained optimisation approaches is that they are typically tailored to specific constraint models (*e.g.*, collision avoidance constraints, or reaching polytopic target sets) with smoothness and continuity assumptions, or specific types of uncertainty (*e.g.*, Gaussian or bounded uncertainty). These restrictive assumptions limit the applicability of these approaches in real-world robotics problems. In contrast, we provide a general framework for chance-constrained finite-horizon optimal control problems with generic chance constraints and generic uncertainty distributions. Building upon the SAA approach, we propose a sample-based approximation that actively accounts for the approximation error caused by the number of samples used in the approximation by adjusting the threshold for constraint violation based on a fixed confidence level. This allows us to use a small number of samples in order to efficiently solve the resulting deterministic problem with the stochastic trajectory optimisation framework VP-STO (Jankowski et al. 2023).

3 Problem Formulation

3.1 Preliminaries

Let Δ be a random variable that models the uncertainty of the system. This can include stochasticity in the dynamics, the environment, and the sensor measurements. A realization $\boldsymbol{\delta}$ of Δ, denoted by $\boldsymbol{\delta} \sim \Delta$, will be referred to as a *sample*, or a *scenario* of the uncertainty.

3.2 Chance-Constrained Optimisation

In the following, we introduce the general chance-constrained optimisation problem. The goal is to find a

solution $\boldsymbol{x}$ that minimizes a cost $J(\boldsymbol{x}) \in \mathbb{R}$ while satisfying a set of constraints. In our work, we consider inequality constraints, *i.e.*, constraints that can be formulated as a function g being negative at $\boldsymbol{x}$, *i.e.*, $g(\boldsymbol{x}) \leq 0$. For instance, g can encode a deterministic collision-avoidance constraint on the robot's distance to obstacles.

Chance-constrained optimisation generalises the above by allowing constraints that depend on a random variable. More precisely, the constraints have the form $g(\boldsymbol{x}, \boldsymbol{\delta}) \leq 0$, where g depends on $\boldsymbol{x}$ and the realization $\boldsymbol{\delta}$ of the uncertainty variable Δ. For instance, $g(\boldsymbol{x}, \boldsymbol{\delta}) \leq 0$ can encode a collision-avoidance requirement of a stochastic system in state $\boldsymbol{x}$ given a particular uncertainty realisation $\boldsymbol{\delta}$. Requiring that $g(\boldsymbol{x}, \boldsymbol{\delta}) \leq 0$ holds for *all* realizations of $\boldsymbol{\delta}$ is often overly conservative, or even infeasible. This is especially true if the distribution of Δ has unbounded support (such as Gaussian noise). Therefore, chance-constrained optimisation relaxes the constraint into a soft constraint, allowing violation of the constraint with a bounded probability η. It thus requires that the probability of a realisation $\boldsymbol{\delta} \sim \Delta$ to satisfy $g(\boldsymbol{x}, \boldsymbol{\delta}) > 0$ is smaller than η. A general chance-constrained optimisation problem can be formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{x} \in X} \quad & J(\boldsymbol{x}) \\ \text{s.t.} \quad & P_{\boldsymbol{\delta} \sim \Delta}[g(\boldsymbol{x}, \boldsymbol{\delta}) > 0] \leq \eta, \end{aligned} \tag{1}$$

where $\boldsymbol{x}$ is the decision variable, constrained in some domain X (e.g., $X \subseteq \mathbb{R}^n$), $J : X \to \mathbb{R}$ is the objective function, and $\eta \in [0, 1]$ is a user-provided threshold for the probability of violating g. We assume that the probability distribution of Δ is known or that we have a generative model for Δ from which we draw samples, *i.e.*, we can draw an arbitrary number of *independent* samples $\boldsymbol{\delta}_1 \sim \Delta, \ldots, \boldsymbol{\delta}_N \sim \Delta$. As mentioned above, the chance constraint is satisfied at $\boldsymbol{x}$ if the probability of violating the constraint g at $\boldsymbol{x}$ is at most η. Computing this probability for a given $\boldsymbol{x}$ is often challenging. For this reason, chance-constrained optimisation problems are often very hard, if not impossible, to solve exactly. To this end, we contribute a tractable approximation of such problems, along with an analysis of the soundness of the approximation.

Remark 1. Note that Eq. (1) can be generalized to multiple chance constraints $P_{\boldsymbol{\delta} \sim \Delta}[g_i(\boldsymbol{x}, \boldsymbol{\delta}) > 0] \leq \eta_i$, for $i = 1, \ldots, L$, with different violation thresholds η_i. However, in this work, we will focus on a single joint constraint ($L = 1$) for simplicity.

Note, that we do not make any additional assumptions on the uncertainty distribution, *i.e.*, it can be of any type and is not restricted to additive noise formulations. However, note that state-dependent uncertainties are outside the scope of this work. The following example illustrates possible sources of uncertainty in a simplified robot motion planning problem.

Example 1. System with uncertain initial condition, actuation noise and uncertain obstacle dynamics. Consider a simple kinodynamic system in discrete time: $s(t+1) = s(t)(u(t) + w(t))$,where $s(t) \in \mathbb{R}$ is the system state, $u(t) \in \mathbb{R}$ the control input, $w(t) \in \mathbb{R}$ the actuation noise, and $z(t) \in \mathbb{R}$ the position of a randomly moving obstacle that occupies the region $[z(t), \infty)$. The random variables $s(0)$, $w(t)$, and $z(t)$ follow known distributions; for instance, $s(0) \sim \mathcal{N}(0, 1)$, $w(t) \sim \mathcal{U}(-1, 1)$, and $z(t) \sim \text{Exp}(\lambda)$ for all t. The objective is to minimize the sum of squared inputs over two time steps, while avoiding the obstacles with high probability over two time steps. Following the formulation of (1), we have $\boldsymbol{x} = (u(0), u(1))$, $\boldsymbol{\delta} = (s(0), w(0), w(1), z(0), z(1), z(2))$, $J(\boldsymbol{x}) = u(0)^2 + u(1)^2$, and $g(\boldsymbol{x}, \boldsymbol{\delta}) = \max_{t=0,1,2} s(t) - z(t)$, where $s(t)$ follows the dynamics introduced above.

3.3 Constraint Satisfaction as a Binary Random Variable

In this subsection, we reformulate the chance constraint in Eq. (1) as a constraint on a binary random variable obtained from Δ. The motivation for doing this is that we will use this formulation to define a *sample average approximation* of the chance constraint in the next section. Concretely, given $\boldsymbol{x}$, we introduce a binary random variable $G_{\boldsymbol{x}} = \mathbf{1}_{g(\boldsymbol{x}, \Delta) > 0}$, wherein $\mathbf{1}_{(\cdot)}$ denotes the indicator function, *i.e.*, for any $\boldsymbol{\delta}$, if $\Delta = \boldsymbol{\delta}$, then $G_{\boldsymbol{x}} = 1$ if $g(\boldsymbol{x}, \boldsymbol{\delta}) > 0$; otherwise, $G_{\boldsymbol{x}} = 0$.

$G_{\boldsymbol{x}}$ is a random variable since it depends on the random uncertainty variable Δ. Thus, we are interested in the probability distribution of the value of $G_{\boldsymbol{x}}$. By definition, this can be obtained from the probability distribution of Δ: namely, $P[G_{\boldsymbol{x}} = 1] = P_{\boldsymbol{\delta} \sim \Delta}[g(\boldsymbol{x}, \boldsymbol{\delta}) > 0]$. Hence, the chance constraint in Eq. (1) can be rewritten as

$$P[G_{\boldsymbol{x}} = 1] \leq \eta. \tag{2}$$

Remark 2. If we know the probability density function p_Δ of Δ, then $P[G_{\boldsymbol{x}} = 1]$ can be obtained by computing the integral

$$P[G_{\boldsymbol{x}} = 1] = \int_{\mathcal{D}} \mathbf{1}_{g(\boldsymbol{x}, \boldsymbol{\delta}) > 0} \, p_\Delta(\boldsymbol{\delta}) \, \mathrm{d}\boldsymbol{\delta}, \tag{3}$$

where the integration domain $\mathcal{D}$ consists of all realizations $\boldsymbol{\delta}$ of Δ. However, computing this integral is generally intractable in practice, especially when the dimension of Δ is large.

4 Sample Average Approximation

Because of the challenges in computing the probability in Eq. (2) exactly (see Remark 2), a tractable approximation is required. A popular approach is the *particle-based* approximation proposed by Blackmore et al. (2010). The core concept is to draw a finite set of i.i.d. uncertainty samples, or samples, and approximate $P[G_{\boldsymbol{x}} = 1]$ as an average over the samples. This approach is justified by the law of large numbers: as the number of samples tends to infinity, the average converges to $P[G_{\boldsymbol{x}} = 1]$. In the remainder of this work, we refer to this as a *sample average approximation* (Pagnoncelli et al. 2009), also known as a Monte Carlo approximation, but we adopt the SAA terminology to highlight its use within an optimisation context.

Formally, consider a set $D = \{\boldsymbol{\delta}_i\}_{i=1}^N$ of N i.i.d. samples drawn from Δ. Based on D, a sample average approximation of $P[G_{\boldsymbol{x}} = 1]$ can be computed as follows:

$$P[G_{\boldsymbol{x}} = 1] \approx \underbrace{\frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_{g(\boldsymbol{x}, \boldsymbol{\delta}_i) > 0}}_{s_N(\boldsymbol{x}; D)}. \tag{4}$$

This is equivalent to counting the number $s_N(\boldsymbol{x}; D)$ of samples $\boldsymbol{\delta}_i$ in $\{\boldsymbol{\delta}_i\}_{i=1}^N$ that violate the constraint g at $\boldsymbol{x}$ and dividing it by the total number of samples N. Note that given $\boldsymbol{x}$ and $\boldsymbol{\delta}$, determining whether $g(\boldsymbol{x}, \boldsymbol{\delta}) \leq 0$ and computing $s_N(\boldsymbol{x}; D)$ for a given solution $\boldsymbol{x}$ is generally much cheaper than computing the integral in Eq. (3).

We can now use the approximation in Eq. (4) to construct a surrogate constraint of the intractable chance constraint in the optimisation problem Eq. (1). A naïve approach is to simply replace $P[G_{\boldsymbol{x}} = 1]$ in Eq. (2) by its approximation, *i.e.*, require that $\frac{1}{N} s_N(\boldsymbol{x}; D) \leq \eta$, or equivalently that $s_N(\boldsymbol{x}; D) \leq \eta N$. By the law of large numbers, when the number of samples approaches infinity, the feasible set of this surrogate constraint asymptotically converges to that of the original chance constraint (cf. Eq. (2)). However, when using a finite number of samples, satisfaction of the surrogate constraint does not guarantee that the original chance constraint is satisfied. The reason is that we need to account for the approximation error in Eq. (4). This can be achieved through strengthening the surrogate constraint: by requiring that

$$s_N(\boldsymbol{x}; D) \leq k_{\text{thresh}} \tag{5}$$

for some $k_{\text{thresh}} < \eta N$. The precise value of k_{thresh} depends on two parameters: *i)* the number of samples N, and *ii)* the level of *confidence* that we want on the *soundness* of the surrogate constraint in Eq. (5). Determining suitable values for k_{thresh} is the main contribution of this paper.

Leveraging Eq. (5), we formulate the following surrogate optimisation problem to the original chance-constrained optimisation problem Eq. (1) as follows:

$$\begin{aligned} \min_{\boldsymbol{x} \in X} \quad & J(\boldsymbol{x}) \\ \text{s.t.} \quad & s_N(\boldsymbol{x}; D) \leq k_{\text{thresh}}. \end{aligned} \tag{6}$$

Our goal is to determine values of k_{thresh} (as a function of N the number of samples) such that feasible solutions of Eq. (6) are feasible for the original problem Eq. (1) with user-given *confidence*.

The term *confidence* above refers to the probability that we sample N samples $D = \{\boldsymbol{\delta}_i\}_{i=1}^N$ for which satisfying the surrogate constraint Eq. (5) implies satisfaction of the original chance constraint in Eq. (2). Although the highest confidence of 1 (100 %) would be desirable, this in general only achievable in the limit, *i.e.*, when $N \to \infty$. Indeed, when N is finite, there is in general a non-zero probability of sampling a set of samples, such that the true chance constraint Eq. (2) may be violated even though the surrogate constraint Eq. (5) is satisfied. However, we can leverage the confidence to establish a connection between the number of samples N and the threshold k_{thresh} in the surrogate constraint Eq. (5) to account for the approximation error arising from the finite number of samples.

4.1 Sample Average Approximation as a Bernoulli Process

The approximation Eq. (4) of $P[G_{\boldsymbol{x}} = 1]$ can be interpreted as a Bernoulli process, *i.e.*, the act of drawing N independent samples from a given binary random variable G. This connection allows us to derive suitable values for k_{thresh} as a function of N and the confidence $1 - \beta$, which we will discuss in the subsequent Secs. 4.2 and 4.3.

Bernoulli process: is a sequence of N i.i.d. binary random variables $G_1, \ldots, G_N$, where each variable follows the same Bernoulli distribution with success probability p, denoted as $G_i \sim \text{Bern}(p)$*. Hence, every variable in the sequence is associated with a Bernoulli trial that has a binary outcome governed by the Bernoulli distribution $\text{Bern}(p)$. The resulting sum of the outcomes of the Bernoulli trials, *i.e.*, $S_N = \sum_{i=1}^N G_i$, is a random variable that follows a *binomial distribution* (Taboga 2017), *i.e.*, for all $k = 0, \ldots, N$,

$$P[S_N = k] = \binom{N}{k} p^k (1-p)^{N-k}, \tag{7}$$

where $p = P[G = 1]$.

In the sample average approximation Eq. (4), the binary random variable G is $G_{\boldsymbol{x}}$ and the corresponding Bernoulli trials are given by $\mathbf{1}_{g(\boldsymbol{x}, \boldsymbol{\delta}_i) > 0}$, $\boldsymbol{\delta}_i \sim \Delta$, for each $i = 1, \ldots, N$. Since with a fixed $\boldsymbol{x}$ and $\{\boldsymbol{\delta}_i\}_{i=1}^N$ being i.i.d., the trials are independent, it holds that for all $k = 0, \ldots, N$,

$$P_{\boldsymbol{\delta}_1 \sim \Delta, \ldots, \boldsymbol{\delta}_N \sim \Delta}[s_N(\boldsymbol{x}; D) = k] = \binom{N}{k} p^\ell (1-p)^{N-k},$$

where $D = \{\boldsymbol{\delta}_i\}_{i=1}^N$ and $p = P[G_{\boldsymbol{x}} = 1]$.

The above yields a closed-form expression of *confidence* through the *cumulative distribution function (CDF)* $C(k; N, p)$ of the binomial distribution with parameters N and p, defined for all $k = 0, \ldots, N$ by

$$C(k; N, p) = \sum_{\ell=0}^{k} \binom{N}{\ell} p^\ell (1-p)^{N-\ell}. \tag{8}$$

4.2 Confidence-Bounded Surrogate Constraint

In the following, we leverage the Bernoulli formulation of the sample average approximation in Eq. (5) and the closed-form expression of the CDF in Eq. (8) to determine a threshold $k_{\text{thresh}} = k_{\text{binom}}(\beta, N, \eta)$†. We set this threshold such that the true chance constraint Eq. (2) is satisfied with a user-defined *confidence* $1 - \beta \in (0, 1)$ (where typically, $\beta \ll 1$). This confidence level applies to any solution $\boldsymbol{x}$ that adheres to the surrogate constraint

$$s_N(\boldsymbol{x}; D) \leq k_{\text{binom}}(\beta, N, \eta). \tag{9}$$

on the sampled set of uncertainty samples $D = \{\boldsymbol{\delta}_i\}_{i=1}^N$. We refer to Eq. (9) as the *confidence-bounded surrogate constraint* to the original chance constraint in Eq. (2). In addition, for simplicity of notation, we also define $\eta_{\text{binom}} = \frac{1}{N} k_{\text{binom}}$.

Proposition 1. *Let $\beta \in (0, 1)$, $N \in \mathbb{N}_{>0}$, $\eta \in (0, 1)$ and let*

$$k_{\text{binom}}(\beta, N, \eta) = \max\left\{k \in \mathbb{N} \mid C(k; N, \eta) \leq \beta\right\} \tag{10}$$

Let $\boldsymbol{x}_{\text{reject}}$ be a solution that violates the chance constraint in Eq. (9), i.e., such that $P[G_{\boldsymbol{x}_{\text{reject}}} = 1] > \eta$. It holds that

$$P_{\boldsymbol{\delta}_1 \sim \Delta, \ldots, \boldsymbol{\delta}_N \sim \Delta}[s_N(\boldsymbol{x}_{\text{reject}}; D) > k_{\text{thresh}}] \geq 1 - \beta, \tag{11}$$

where $D = \{\boldsymbol{\delta}_i\}_{i=1}^N$ and $k_{\text{thresh}} = k_{\text{binom}}(\beta, N, \eta)$.

---

*Note that in the more general definition, the sequence of a Bernoulli process can also be infinite.

†When clear from the context, we will drop the arguments β, N and η in $k_{\text{binom}}(\beta, N, \eta)$.

The inequality in Eq. (11) is a lower bound on the probability of *correctly rejecting* a candidate solution $\boldsymbol{x}$ by means of the surrogate constraint in Eq. (9).

Proof. Given $\boldsymbol{x}_{\text{reject}}$ as in the proposition, we look at the probability that $s_N(\boldsymbol{x}_{\text{reject}}; D) \leq k_{\text{thresh}}$, *i.e.*, the probability that we do not reject $\boldsymbol{x}_{\text{reject}}$ when using the surrogate constraint in Eq. (9). Since $G_{\boldsymbol{x}_{\text{reject}}}$ is a binary random variable with probability $p = P[G_{\boldsymbol{x}_{\text{reject}}} = 1]$ and $D = \{\boldsymbol{\delta}_i\}_{i=1}^N$ are independent, the sum $s_N(\boldsymbol{x}_{\text{reject}}; D)$ follows a binomial distribution with parameters N and p, for which the CDF is given by Eq. (8). This implies that the probability that $s_N(\boldsymbol{x}_{\text{reject}}; D) \leq k_{\text{binom}}$ is equal to $C(k_{\text{binom}}; N, p)$. We build on the fact that the binomial distribution is monotonic with respect to p (Taboga 2017), *i.e.*, $p_1 < p_2$ implies $C(k; N, p_1) > C(k; N, p_2)$. Hence, it holds that $P_{\boldsymbol{\delta}_1 \sim \Delta, \ldots, \boldsymbol{\delta}_N \sim \Delta}[s_N(\boldsymbol{x}_{\text{reject}}; D) \leq k_{\text{binom}}]$ is smaller than or equal to $C(k_{\text{binom}}; N, \eta)$ since $p > \eta$. Now, by definition of k_{binom}, it holds that $C(k_{\text{binom}}; N, \eta) \leq \beta$, so that the probability of not rejecting $\boldsymbol{x}_{\text{reject}}$ is at most β.

In Fig. 2, we show the CDF of the binomial distribution for different values of N and p. From the plots, we can obtain $k_{\text{binom}}(\beta, N, p)$ by looking at the intersection of the CDF with the horizontal line $y \equiv \beta$. The key insight from the derivation of the above framework is that using the naïve SAA approach, *i.e.*, $k_{\text{thresh}} = \eta N$, corresponds to a confidence $\beta \approx 0.5$. Indeed, in Fig. 2, we can see that $k_{\text{binom}}(0.5, N, p) \approx pN$ because the horizontal line $y \equiv 0.5$ intersects the curves roughly at $k/N = p$. This highlights a key limitation of the naïve approach: by implicitly operating at a confidence level of around 50%, it tends to accept solutions that have a high chance of violating the true constraint, despite appearing feasible on the sampled data. This is further illustrated in Sec. 11 and Fig. 13 in the Appendix.

The idea of expressing a chance constraint in terms of a sample-based variable $s_N(\boldsymbol{x}; D)$ and the inverse CDF, *i.e.*, $\text{CDF}^{-1}(\beta)$ is not new; see, *e.g.*, Heirung et al. (2018); Peña-Ordieres et al. (2020). In fact, it has been used in the past to derive theoretical bounds on the number of samples needed to ensure constraint satisfaction in scenario optimisation approaches; see, *e.g.*, Campi and Garatti (2011). Yet, to the best of our knowledge, it has never been used within the Boolean formulation of a chance constraint and its sample average approximation. Instead, it has only been used for simple continuous constraints with tractable distributions for which the CDF could be derived analytically.

4.3 Limitation of the Approximation

In summary, the confidence-bounded surrogate constraint Eq. (9) is based on the formulation of the SAA as a Bernoulli process. However, a crucial assumption in Proposition 1 is that the Bernoulli variables $G_{\boldsymbol{x},i} = \mathbf{1}_{g(\boldsymbol{x},\boldsymbol{\delta}_i)>0}$ for $i = 1, \ldots, N$, are independent, such that the sum of the Bernoulli variables $s_N(\boldsymbol{x}; D)$ follows a binomial distribution. However, when applied to the output of (6), although the samples $\{\boldsymbol{\delta}_i\}_{i=1}^N$ themselves are independent, the Bernoulli variables $G_{\boldsymbol{x},i}$ are generally not independent, as the solution $\boldsymbol{x}$ depends on these samples through the optimisation scheme. Hence, only if $\boldsymbol{x}$ is independent from the samples $\{\boldsymbol{\delta}_i\}_{i=1}^N$, does $s_N(\boldsymbol{x}; D)$ follow a binomial distribution, and Proposition 1 holds. Said otherwise, Proposition 1 gives the probability of rejecting a *given* infeasible solution $\boldsymbol{x}_{\text{reject}}$, independent of the samples. Nevertheless, the probability of rejecting *all* infeasible solutions is in general strictly smaller. Consequently, without additional assumptions that ensure independence, solving the surrogate optimisation problem Eq. (6) with the confidence-bounded threshold k_{binom} is a *heuristic* approach.

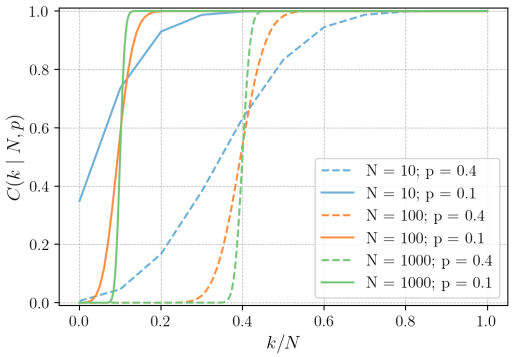


Figure 2. CDF of the binomial distribution for different values of N and p, given k/N on the x-axis. The CDF values map to our confidence in observing k/N constraint violations under the assumption that the true probability of constraint violation is p.

4.4 Addressing the Limitation

In the following, we present two settings where the optimisation framework (6) can be applied under reasonable independence assumptions. These represent cases where our heuristic approach offers *firm guarantees*. For the second setting, this also provides insight into why the surrogate performs well in practice, even if the underlying assumption is difficult to verify in practice, or may be satisfied only part of the time.

4.4.1 A-posteriori validation In the first setting, Eq. (6) is used as an *a-posteriori* validation step. This works as follows: Given a candidate solution $\hat{\boldsymbol{x}}$, *e.g.*, obtained by solving Eq. (6) using a sample set $\hat{D}$, we re-evaluate the constraint function g at $\hat{\boldsymbol{x}}$ using a new set of N i.i.d. samples $D = \{\boldsymbol{\delta}_i\}_{i=1}^N$. If the empirical violation count $s_N(\hat{\boldsymbol{x}}; D)$ exceeds the acceptance threshold k_{binom}, the candidate solution is rejected.

Proposition 2. (Informal). *If the candidate solution passes the validation test, then with confidence $1 - \beta$ it satisfies the true chance constraint.*

Proof. This setup falls within the scope of Proposition 1, since $\hat{\boldsymbol{x}}$ and the validation set D are independent. As a result, if $\hat{\boldsymbol{x}}$ violates the true chance constraint, Proposition 1 guarantees that it will be rejected with probability at least $1 - \beta$ during the *a-posteriori* validation. Conversely, if the candidate passes validation, we can assert with confidence at least $1 - \beta$ that it satisfies the chance constraint.

However, this raises the question of how to proceed when a candidate solution is rejected. One option is to repeat the optimisation and validation steps until a

candidate passes. Yet, doing so introduces a *multiple hypothesis testing problem*: although each individual test maintains a confidence level of $1-\beta$, the *overall* probability of accepting a violating solution increases unless this accumulation is properly corrected. Statistical correction methods, such as those discussed by Abdi et al. (2007), can mitigate this issue, though at the cost of increased conservatism or a higher required sample size. A more rigorous analysis of *a-posteriori* validation as a verification mechanism for certifying the safety and performance of robotic policies is provided in Vincent et al. (2024).

In time-sensitive applications such as online planning, performing multiple rounds of optimisation and validation, potentially with increasing sample counts, may not be feasible. In such cases, it is advisable to cap the number of candidate evaluations (or limit the computation time) and instead rely on fallback strategies or recovery controllers to ensure constraint satisfaction when no validated solution is found.

4.4.2 Receding-horizon optimisation The second setting where our approach can offer firm guarantees is the receding-horizon MPC context, where Eq. (6) is solved repeatedly at each control step. We provide an assumption (Assumption 1) under which a form of independence of the Bernoulli variables holds. While this assumption may not strictly hold at every step or be directly verifiable in practice, it provides a plausible theoretical justification for the empirical effectiveness of our approach in the MPC setting. In other words, even though our method is heuristic in the general setting, this setting gives a context in which it behaves reliably and aligns with our experimental observations (see Sec. 6.2.3). We now formalize this intuition with an assumption and proposition:

At each MPC step $m = 1, \dots, M$, we compute a new solution $\boldsymbol{x}_m$ with a new sample set $D = \{\boldsymbol{\delta}_{m,i}\}_{i=1}^{N}$. That solution is executed for a few milliseconds, until we re-sample and compute a new solution in the next MPC step. In general, the solution $\boldsymbol{x}_m$ is *not very different* from the solution $\boldsymbol{x}_{m-1}$ (if needed, this can also be enforced as an explicit constraint of the MPC). In this case, provided the constraint function g varies smoothly with $\boldsymbol{x}$, we can make the assumption that the binary trials $\mathbf{1}_{g(\boldsymbol{x}_m, \boldsymbol{\delta}_{m,i})>0}$ are independent because $\{\boldsymbol{\delta}_{m,i}\}_{i=1}^{N}$ is i.i.d. and $\boldsymbol{x}_m \approx \boldsymbol{x}_{m-1}$. We formalize this with an assumption and a proposition:

Assumption 1. There is $\epsilon > 0$ such that with probability $1-\epsilon$ on $\boldsymbol{\delta}$, if there is $\boldsymbol{x} \in X$ such that $g(\boldsymbol{x}, \boldsymbol{\delta}) > 0$, then for all $\boldsymbol{x} \in X$, it holds that $g(\boldsymbol{x}, \boldsymbol{\delta}) > 0$.

See Fig. 3 for an illustration of Assumption 1.

Proposition 3. *Under Assumption 1, it holds that*

$$P_{\boldsymbol{\delta}_1 \sim \Delta, \dots, \boldsymbol{\delta}_N \sim \Delta}[P[G_{\boldsymbol{x}} = 1] \leq \eta + \epsilon] \geq 1 - \beta, \tag{12}$$

where $\boldsymbol{x}$ is the solution of the surrogate optimisation problem in Eq. (6) with $D = \{\boldsymbol{\delta}_i\}_{i=1}^{N}$ and $k_{\text{thresh}} = k_{\text{binom}}(\beta, N, \eta)$.

Proof. Let $\boldsymbol{x}$ be the solution of Eq. (6) with $D = \{\boldsymbol{\delta}_i\}_{i=1}^{N}$. Assume that $P[G_{\boldsymbol{x}} = 1] > \eta + \epsilon$. By Assumption 1, this implies that $P_{\boldsymbol{\delta} \sim \Delta}[\min_{\boldsymbol{x}' \in X} g(\boldsymbol{x}', \boldsymbol{\delta}) > 0] > \eta$. Furthermore,

$$s_N^{\min}(D) \triangleq \sum_{i=1}^{N} \min_{\boldsymbol{x}' \in X} \mathbf{1}_{g(\boldsymbol{x}', \boldsymbol{\delta}_i)>0} \leq s_N(\boldsymbol{x}; D) \leq k_{\text{binom}}.$$

Since the variables $\min_{\boldsymbol{x}' \in X} \mathbf{1}_{g(\boldsymbol{x}', \boldsymbol{\delta}_i)>0}$ for $i = 1, \dots, N$, are i.i.d., it follows that $s_N^{\min}(D)$ is a Bernoulli process with N trials and $p = P_{\boldsymbol{\delta} \sim \Delta}[\min_{\boldsymbol{x}' \in X} g(\boldsymbol{x}', \boldsymbol{\delta}_i) > 0]$. Hence, the probability that $s_N^{\min}(D) \leq k_{\text{binom}}$ is equal to $C(k_{\text{binom}}; N, p)$. The rest follows in the same way as in the proof of Proposition 1.

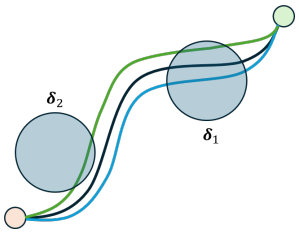


Figure 3. When $\boldsymbol{\delta} = \boldsymbol{\delta}_1$, the obstacle collides with all three paths, whereas when $\boldsymbol{\delta} = \boldsymbol{\delta}_2$, the obstacle collides only with the green path. Assumption 1 states that $\boldsymbol{\delta}$ takes a value for which a situation like $\boldsymbol{\delta}_2$ occurs with probability at most ϵ.

Remark 3. In a receding-horizon optimisation scheme, under the assumption that the solution does not vary too much from one step to the next one, the samples used at the previous steps also provide indication that the solution at the current step is valid. This information is *not* used in the confidence bound that we can derive from Proposition 3, since it is for a single MPC step. This is why in practice the real value of $P[G_{\boldsymbol{x}_m} = 1]$ is often smaller than η, as we will show in the experiments (see Sec. 6). In future work, we plan to use and quantify this information to obtain even less conservative bounds on $P[G_{\boldsymbol{x}_m} = 1]$.

4.5 Relationship between Chance Constraints and Conditional Value-at-Risk

In the following, we introduce the concept of *conditional value-at-risk* (CVaR) (Majumdar and Pavone 2020) and its relationship to chance constraints in the context of binary indicator functions. This relationship will be helpful to understand subsequent results comparing the two formulations. In contrast to formulating chance constraints, the concept of constraining the CVaR depends on the topology of the constraint function g with respect to the realization of the disturbance. In general, Lew et al. (2023) shows that using a chance constraint as in Eq. (1) is equivalent to constraining the *value-at-risk* (VaR) with

$$\text{VaR}_\eta(g(\boldsymbol{x}, \boldsymbol{\delta})) = \inf_{\lambda \in \mathbb{R}} \{\lambda \mid P_{\boldsymbol{\delta} \sim \Delta}[g(\boldsymbol{x}, \boldsymbol{\delta}) > \lambda] \leq \eta\} \leq 0. \tag{13}$$

Furthermore, it can be shown that constraining the CVaR is strictly more conservative, *i.e.*, Eq. (13) holds if $\text{CVaR} \leq 0$ (Lew et al. 2023).

In the following, we show that constraining the CVaR of a binary indicator function corresponds to a tighter chance constraint with a lower effective chance threshold. As the

CVaR corresponds to the expected value of the constraint function for $g > \text{VaR}$, the CVaR of a binary indicator function can be reformulated in terms of the probability of violating the constraint g, *i.e.*,

$$\text{CVaR}_\eta(G_{\boldsymbol{x}}) = \begin{cases} \frac{P[G_{\boldsymbol{x}}=1]}{\eta}, & \text{if} P[G_{\boldsymbol{x}}=1] < \eta, \\ 1, & \text{otherwise.} \end{cases} \tag{14}$$

The CVaR_η of the binary indicator function in Eq. (14) is in the range $[0, 1]$. Next, we may define a threshold $\text{CVaR}_{\max} \in [0, 1]$ in order to construct a constraint based on the CVaR with $\text{CVaR}(G_{\boldsymbol{x}}) \leq \text{CVaR}_{\max}$. By using Eq. (14), it follows that constraining the CVaR yields another threshold on the probability of violating the constraint, *i.e.*,

$$P[G_{\boldsymbol{x}} = 1] \leq \eta\, \text{CVaR}_{\max} \iff \text{CVaR}_\eta(G_{\boldsymbol{x}}) \leq \text{CVaR}_{\max}. \tag{15}$$

Note that for any $\text{CVaR}_{\max} \in [0, 1)$, the resulting constraint is more conservative than the VaR constraint in Eq. (13).

This increased conservatism has practical implications in control and planning under uncertainty. In particular, CVaR-based constraints offer stronger safety guarantees by effectively lowering the allowable probability of constraint violation. However, this safety margin comes at the cost of increased conservatism, which may lead to overly cautious or suboptimal behavior in less risk-sensitive settings. Therefore, understanding the trade-off between chance constraints and CVaR constraints is essential for selecting the appropriate level of risk aversion based on the application's requirements, as we will show in Sec. 6.

5 Stochastic Trajectory Optimisation with Chance Constraints

Due to the non-smooth nature of the uncertainty dynamics and the resulting non-smooth surrogate chance constraint with respect to the optimisation variable in Eq. (9), we approach the optimisation problem with a gradient-free, *i.e.*, zero-order, evolutionary optimisation technique. Building upon our previous work *Via-Point-Based Stochastic Trajectory Optimisation (VP-STO)* (Jankowski et al. 2023), we introduce *chance-constrained VP-STO (CC-VPSTO)* for finding robot trajectories that minimise a given task-related objective *while satisfying a given chance constraint*.

5.1 Preliminaries on VP-STO

VP-STO builds on stochastic optimisation in order to find robot trajectories that minimise a given task-related objective in dynamic environments.

Trajectory Representation In VP-STO the decision variable $\boldsymbol{x}$ for an optimisation problem, such as the one in Eq. (1), is a set of S via-points $\boldsymbol{q}_{\text{via}} = (q_{\text{via},1}, \ldots, q_{\text{via},s})$, *i.e.*, $\boldsymbol{x} = \boldsymbol{q}_{\text{via}}$. For a given set of via-points, VP-STO synthesises a time-continuous and smooth trajectory that satisfies the boundary conditions, such as initial and final state and velocity, and kinodynamic constraints, such as velocity and acceleration limits[‡]. The advantage of the approach lies in the low-dimensional representation of the trajectory, which allows for efficient optimisation in a low-dimensional space.

Optimisation Algorithm VP-STO uses the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) (Hansen 2016) to find the optimal set of via-points that map to a trajectory that minimises the given objective function. CMA-ES iteratively updates the mean and covariance of a Gaussian distribution that represents the search space of the optimisation problem, *i.e.*, the set of via-points. In each iteration j, the algorithm samples candidate solutions from this distribution, *i.e.*, $\mathcal{N}\left({}^j\boldsymbol{\mu}_{\text{via}}, {}^j\boldsymbol{\Sigma}_{\text{via}}\right)$, evaluates them on the given objective function, and updates the distribution based on the evaluation results. The algorithm converges to the optimal solution in a few iterations, making it suitable for real-time applications.

5.2 Chance-Constrained VP-STO

VP-STO has been shown to be effective in generating robot trajectories in real-time for dynamic environments, outperforming state-of-the-art sampling-based MPC methods (Bhardwaj et al. 2022). Yet, in its original form, VP-STO does not consider uncertainty, but instead assumes a deterministic environment. In this work, we extend the VP-STO framework to consider uncertainty in the environment, *i.e.*, we introduce the *chance-constrained VP-STO (CC-VPSTO)* framework.

CC-VPSTO solves the optimisation problem in (6) using the surrogate constraint in Eq. (9). We enforce the constraint through a penalty-based approach, *i.e.*, we include the constraint $k \leq k_{\text{thresh}}$ in the objective function as a penalty term. This can be seen as a discontinuous barrier function that adds a very high penalty term J_{pen} to the objective function if the constraint is violated, *i.e.*, when the observed number of constraint violations $k > k_{\text{thresh}}$. The closed-form formulation of this penalty term is as follows:

$$J_{\text{pen}} = \mathbf{1}[k > k_{\text{thresh}}] \cdot (J_{\text{pen,min}} + a \cdot (k - k_{\text{thresh}} - 1)).$$

We choose the minimum penalty term $J_{\text{pen,min}}$ to be much larger than the maximum cost objective without constraint violations. Moreover, we add a piecewise linear term to the minimum penalty term that grows linearly with the extra number of violations compared to k_{thresh}. This term makes the constraint landscape smoother and gives the optimiser a direction towards feasible solutions without violations. The overall algorithm for CC-VPSTO is summarised in Algorithm 1, where the approximation of the chance constraint, *i.e.*, counting the number of samples that cause the solution to violate the constraint, is encapsulated in the `evaluate` function.

In our previous work, we demonstrated the suitability of the VP-STO framework for real-time robot motion planning in dynamic environments (Jankowski et al. 2023). Similarly, the CC-VPSTO framework, *i.e.*, the above algorithm, can be used in a receding horizon MPC scheme to generate robot trajectories in real-time. Yet, we note that the constraint evaluation in the `evaluate` function will be computationally more expensive than in the original VP-STO framework, as it requires N Monte Carlo simulations

[‡]For more information on how we generate the continuous trajectories from via-points, we refer the reader to Sec. 9 in the Appendix and to (Jankowski et al. 2023, 2022).

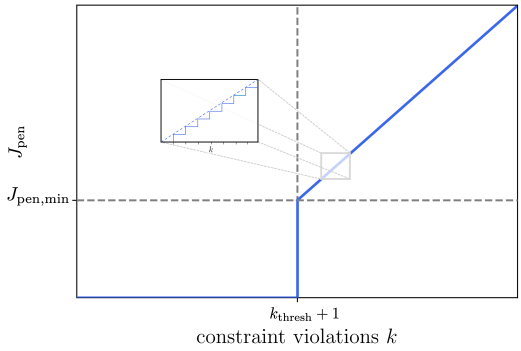


Figure 4. Graph of the penalty function used in CC-VPSTO. When observing more than k_{thresh} constraint violations in the N Monte Carlo simulations, the penalty function takes value $J_{\text{pen,min}}$ plus a quantity proportional to the number of extra constraint violations compared to k_{thresh}. Note that we chose the minimum penalty term $J_{\text{pen,min}}$ to be much larger than the largest cost objective without constraint violations.

per candidate trajectory $\boldsymbol{\xi}$. This implies that the reactivity of our framework now depends on the number of samples N used in the approximation. The advantage of our approximation is that it allows us to choose the number of samples and then use confidence levels to determine an appropriate threshold. Crucially, the number of samples can be selected based on the target execution frequency of the MPC scheme, enabling a trade-off between computational efficiency and approximation accuracy. In contrast, scenario-based optimisation typically requires a large number of samples, derived from conservative theoretical bounds that do not account for real-time constraints. This limits its practical applicability in time-sensitive settings. Last, we note, that VP-STO and thus also CC-VPSTO in the current form does not handle non-holonomic constraints, such as those arising from differential drive robots, which we leave for future work.

6 Experiments

We evaluate our framework, *i.e.*, Algorithm 1, with and without the MPC scheme, in simulations and in a real-world experiment with a Franka Emika robot arm. The simulation experiments allow us to make claims about the empirical performance of our system across different settings and parameterisations. The robot experiment allows us to evaluate the real-time applicability of our approach. The supplementary video includes videos from both simulated and real experiments. These can also be found on our website https://sites.google.com/oxfordrobotics.institute/cc-vpsto.

6.1 Experimental Setup

6.1.1 Joint probability of constraint violation In all our experiments the chance constraint is formulated on the collision probability with obstacles in the robot's environment. We encode this as a *joint* chance constraint, *i.e.*, enforcing *trajectory-wise* constraint satisfaction with high probability (cf. Sec. 2 for more details). A joint formulation is more meaningful interpretation for robot behaviour, in

Algorithm 1: CC-VPSTO

Input: $\boldsymbol{q}_0, \dot{\boldsymbol{q}}_0, \boldsymbol{q}_T, \dot{\boldsymbol{q}}_T, \dot{\boldsymbol{q}}_{\min}, \dot{\boldsymbol{q}}_{\max}, \ddot{\boldsymbol{q}}_{\min}, \ddot{\boldsymbol{q}}_{\max}, N_{via}$, $maxIter, S, H, \eta, \beta, N$

/* N_{via}: no. of via-points, */
/* $maxIter$: max. no. of CMA-ES iterations, */
/* S: no. of sampled candidate trajs. */
/* H: horizon */
/* η: chance constraint threshold */
/* β: confidence threshold */
/* N: no. of samples */

Output: Robot trajectory $\boldsymbol{\xi}^*_{0:H}$

${}^0\boldsymbol{\mu}_{\text{via}}, {}^0\boldsymbol{\Sigma}_{\text{via}} \leftarrow \text{init}(N_{via})$
$j \leftarrow 0$
Sample $\Delta \leftarrow \{\boldsymbol{\delta}_i \sim p_\Delta\}_{i=1}^N$
$k_\beta \leftarrow k_{\text{binom}}(\beta, N, \eta)$
while $j < maxIter$ **do**
  $\{\boldsymbol{q}_{\text{via}}\}_{s=1}^S \leftarrow \text{sample}\big({}^j\boldsymbol{\mu}_{\text{via}}, {}^j\boldsymbol{\Sigma}_{\text{via}}\big)$ // via-points
  $\{\boldsymbol{\xi}\}_{s=1}^S \leftarrow \text{synthesise}\big(\{\boldsymbol{q}_{\text{via}}\}_{s=1}^S\big)$ // trajectories
  $\{c\}_{s=1}^S \leftarrow \text{evaluate}\big(\{\boldsymbol{\xi}\}_{s=1}^S, k_\beta, \Delta\big)$ // cost
  ${}^{j+1}\boldsymbol{\mu}_{\text{via}}, {}^{j+1}\boldsymbol{\Sigma}_{\text{via}} \leftarrow \text{CMA-ES}\big(\{\boldsymbol{q}_{\text{via}}, c\}_{s=1}^S\big)$
  $j \leftarrow j+1$
end
$\boldsymbol{\xi}^*_{0:H} \leftarrow \text{synthesise}\big(\boldsymbol{\mu}^j_{\text{via}}\big)$

contrast to evaluating the constraint independently at each time step. This means that we would not consider a trajectory to be safe if it avoids collisions in one time step with a very high probability, but collides in the next time step. We thus consider correlation over time in the chance constraint, *i.e.*, the first collision in a trajectory renders the whole trajectory unsafe and all subsequent collisions do not add any additional risk (Lew et al. 2023).

6.1.2 Uncertainty Samples Before outlining our experiments in detail, we first clarify the role of uncertainty samples in both *i)* our algorithm and *ii)* its evaluation. In each of the experiments, we draw separate sample sets for the optimisation and for reporting satisfaction of the chance constraint within the experiment. Across all experiments, an uncertainty sample corresponds to a single possible realisation of how the environment may evolve, in other words, a scenario. a sample represents a specific obstacle position drawn from a distribution (*e.g.*, a Gaussian). When multiple dynamic obstacles are present, a single sample consists of M predicted trajectories, one for each of the M obstacles. The uncertainty distribution used for both optimisation and evaluation is assumed to be the same and fixed throughout the experiment.

6.1.3 Collisions In the case of a single obstacle, we consider a robot trajectory to be in collision if the robot collides with the obstacle at *any point in time* across the whole trajectory. For multiple obstacles, we extend this definition to say a robot trajectory is in collision if the robot would collide with *any* obstacle *at any point in time*. By this, we avoid double counting collisions. One uncertainty sample can only be counted as one collision, even in cases where it might collide with several obstacles at different points in time.

6.2 Simulation Experiments

All simulation experiments are conducted in a bounded 2D environment with a circular holonomic robot, as shown in Fig. 6. In Sec. 6.2.1, we perform offline planning experiments where CC-VPSTO is run once to compute a trajectory over the full horizon from start to goal under Gaussian uncertainty. In Sec. 6.2.2, we evaluate the same offline setting but with multi-modal, non-Gaussian uncertainty to demonstrate the method's flexibility beyond standard distributional assumptions. Last, in Sec. 6.2.3 we evaluate the online planning case, where we follow a receding horizon approach using CC-VPSTO to re-plan the trajectory at every MPC step. The results of the offline experiments will also be relevant for the online setting, as each online replanning step can be seen as solving a new offline optimisation problem. In both experiment settings, the uncertainty stems from obstacles in the environment, with no uncertainty in the robot dynamics[§]. In the offline planning setting, obstacles are static but have uncertain positions, modeling the effect of measurement noise. In contrast, in the online receding-horizon setting, obstacles dynamically move according to a random walk model. Their velocities are reversed upon hitting workspace boundaries, keeping them within workspace bounds and introducing non-linear dynamics. In all experiments, the obstacles are circular with varying radii, but our optimisation scheme does not rely on convexity and can accommodate more complex obstacle shapes.

6.2.1 Offline Planning (Gaussian Uncertainty) In this offline planning setting, we show the properties of CC-VPSTO with a single static obstacle whose uncertain position follows a Gaussian distribution, as shown in Fig. 6 (see Sec. 6.2.2 for results on a non-Gaussian distribution). Here, a sample of the uncertainty refers to a possible (static) position of the obstacle, as explained in more detail in Sec. 6.1.

For every combination of values of N (100, 1000), η (0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.6, 0.8) and $\beta = 0.05$, we run $N_{\text{exp}} = 10^5$ experiments. For $i = 0, \ldots, N_{\text{exp}}$ sample sets of N i.i.d. samples, we compute the trajectory ξ_i using CC-VPSTO with $k_{\text{binom}}(\beta, \eta, N)$. Then, for each trajectory ξ_i, we evaluate its probability $\hat{\eta}_i$ of colliding with the static uncertain obstacle as follows. We use a new set of $N_{\text{eval}} = 10^4$ i.i.d. samples, sampled from the same distribution of possible obstacle locations, and count the number of samples that collide with ξ_i. The ratio of this number by N_{eval} is $\hat{\eta}_i$. We use $\hat{\eta}_i$ to compute the following three metrics:

1. ***Mean collision probability*:**
   $\hat{\eta}_{\text{avg}} = \sum_{i=0}^{N_{\text{exp}}} \hat{\eta}_i / N_{\text{exp}}$
2. **$(1-\beta)$*-percentile of the collision probability*:**
   $\hat{\eta}_{(1-\beta)} = \text{percentile}(\{\hat{\eta}_i\}_{i=0}^{N_{\text{exp}}}, 1-\beta)$
3. ***Probability of chance constraint violation*:**
   $P(\hat{\eta}_i > \eta) = \hat{\beta} = \left(\sum_{i=0}^{N_{\text{exp}}} \mathbf{1}_{\hat{\eta}_i > \eta}\right) / N_{\text{exp}}$

Note that we denote *empirical* values with a hat, *e.g.*, $\hat{\eta}$. For the proposed heuristic bound to be a good approximation, a proportion of maximum β of the solutions can be in collision. This is because we set our confidence threshold to $1-\beta$ A trajectory ξ_i violates the chance constraint if its estimated value $\hat{\eta}_i$ exceeds η.

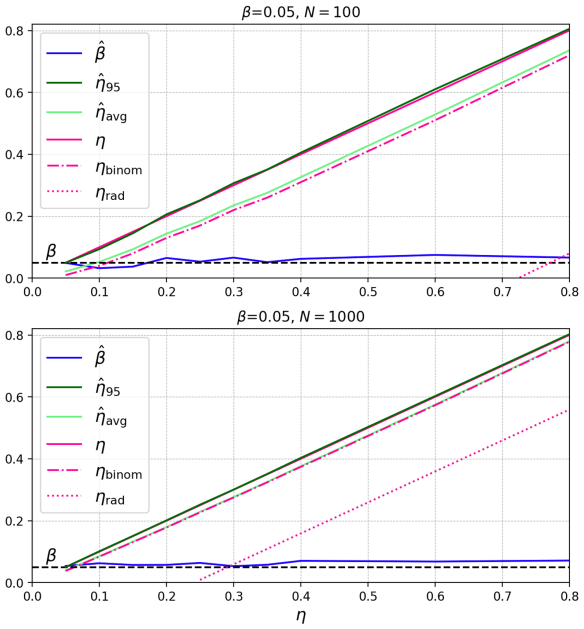


Figure 5. ***Offline Planning Experiment.*** We evaluate the proposed binomial bound η_{binom} in a Gaussian offline-planning setting by running CC-VPSTO $N_{\text{exp}} = 10^5$ times for different risk levels η and sample budgets $N \in \{100, 1000\}$. Each resulting trajectory is then assessed on a new set of $N_{\text{eval}} = 10^4$ unseen obstacle samples to estimate its empirical collision probability $\hat{\eta}_i$. We report three aggregate metrics across experiments: the *mean collision probability* $\hat{\eta}_{\text{avg}}$, the 95*-percentile* $\hat{\eta}_{95}$, and the *empirical chance-constraint violation rate* $\hat{\beta}$ (fraction of runs with $\hat{\eta}_i > \eta$). Theoretical curves for η_{binom} and the Rademacher-based baseline η_{rad} are shown for comparison. Importantly, the binomial bound η_{binom} (magenta, dash–dot) consistently provides a tight and accurate approximation of the true collision probabilities, especially for lower sample counts N.

Baseline In the course of this work, we developed an alternative approach to approximate the chance constraint in Eq. (1) based on the *Rademacher complexity* from statistical learning theory (Shalev-Shwartz and Ben-David 2014; Mohri et al. 2018). Computing a suitable k_{thresh} for the surrogate optimisation problem in Eq. (6) can be approached by computing an upper bound on the Rademacher complexity of the associated set of functions. However, despite the theoretical attractivity of the this approach, computing an upper bound on the Rademacher complexity can be very challenging in general, and there is usually no closed-from expression for such bounds. Yet, we found a tight bound k_{rad} for a special case of collision-avoidance problem. This bound does not require the independence of the Bernoulli variables, but it is more conservative, computationally expensive, and, less general since it is limited to a specific motion planning problem. Consequently, we use this bound as a baseline for our

[§]This is for simplicity only. The extension to process noise and external disturbances is straightforward given the proposed approach.

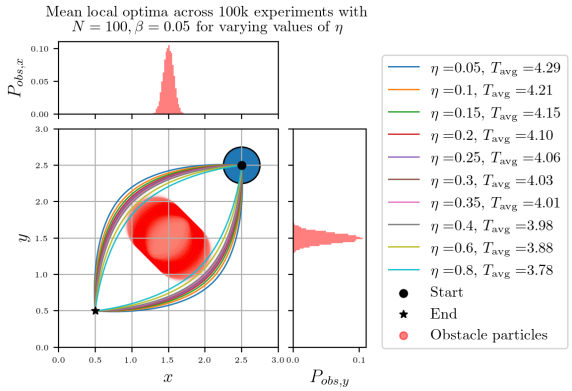


Figure 6. ***Offline Planning Experiment (Gaussian Uncertainty).*** We show $N_{\text{eval}} = 10^4$ red circles for the uncertain obstacle position and the mean trajectory for the two local optima from CC-VPSTO, which used $N = 100$ samples in the optimisation, for varying values of η across $N_{\text{exp}} = 10^5$ experiments. The blue circle shows the robot's radius and starting position.

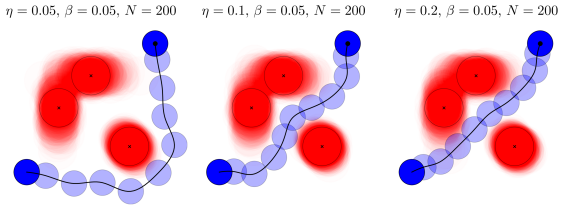


Figure 7. ***Offline Planning Experiment (Multimodal Uncertainty).*** We show $N_{\text{eval}} = 10^4$ red circles for the uncertain obstacle position. The black crosses indicate the means of the three Gaussian modes. The black trajectory and the blue circles illustrate the optimal trajectory computed with CC-VPSTO, which used $N = 200$ samples in the optimisation, for varying values of η.

Table 1. *Results of the Offline Planning Experiment (Multimodal Uncertainty).*

	$\eta = 0.05$	$\eta = 0.1$	$\eta = 0.2$
$\hat{\eta}_{\text{avg}}$	0.026	0.086	0.164
$\hat{\beta}\,(\beta = 0.05)$	0.026	0.228	0.069
T_{avg}	4.702	3.164	2.752

simulation experiments. The full derivation, the closed-form expression for k_{rad}, and the proofs and assumptions can be found in Appendix 10.

Results The results of our offline analysis are summarised in Fig. 5. The pink dotted curves show the theoretical values of η_{binom} and η_{rad}, and the green curves show the empirical values of $\hat{\eta}_{\text{avg}}$ and $\hat{\eta}_{(1-\beta)}$ for CC-VPSTO. In addition, we plot the empirical probability of chance constraint violation $\hat{\beta}$ in blue against the user-defined value of β. Note that we also provide the exact numerical results from Fig. 5 in Tab. 2 in the Appendix. We observe that our proposed bound k_{binom} provides a sufficient value for k_{thresh} since the observed $\hat{\eta}_{0.95}$ is close to the target η (or equivalently, $\hat{\beta}$ is close to β). This means that empirically the probability of collision is below η, with confidence $95\,\%$. We also observe that when N is larger, η_{binom} and $\hat{\eta}_{\text{avg}}$ are closer to η, implying that the surrogate optimisation problem becomes less conservative as the number of samples increases, given the same user-defined confidence-level. This is expected since more samples provide a better approximation of the distribution; at the cost of increased computation time. Last, the figure also shows that the baseline bound η_{rad} is much more conservative, as it is significantly smaller than η_{binom}. Moreover, the offset from η_{binom} increases substantially when decreasing the number of samples N. In addition to the quantitative results, we visualize the mean trajectories for the two local optima found by CC-VPSTO for different values of η in Fig. 6. Note, that we only show the solutions for the experiments with $N = 100$ in the optimisation, as the solutions for $N = 1000$ are visually indistinguishable. In the legend of Fig. 6, we also show the average motion duration of the trajectories across experiments for the different values of η. This qualitative analysis shows that with higher values of η, CC-VPSTO finds more efficient, but also less conservative trajectories, as the mean trajectories are closer to the obstacle, since they allow for a higher probability of collision.

6.2.2 Offline Planning (Multimodal Uncertainty) CC-VPSTO does not make any assumptions about the uncertainty distribution and is able to handle arbitrary distributions. To illustrate this, we provide an additional offline motion planning experiment where we replace the Gaussian distribution over the obstacle position in the previous experiment from Sec. 6.2.1 with a Gaussian mixture distribution with three modes. Fig. 7 illustrates the qualitative results for this scenario with a multimodal distribution. It shows that CC-VPSTO is able to find a timing-optimal trajectory given the non-Gaussian uncertainty over the obstacle position depending on the user-defined chance threshold η and confidence threshold $1 - \beta$. Yet, note that a more complex distribution as the one given by the Gaussian mixture distribution by default requires more samples in the approximation. This is because multi-modal distributions exhibit higher variance and can contain isolated high-probability regions that are easily missed with insufficient sampling. Capturing the shape and support of such distributions reliably in the sample-based surrogate requires a denser sampling of the space, which is why we used $N = 200$ samples in this experiment. Table 1 provides numerical results on the average probability of collision of the solution ($\hat{\eta}_{\text{avg}}$), the empiric estimation of β, and the average duration of the solution trajectory T (i.e. the optimisation objective). These statistical results are computed from 1000 experiments performed for each η and evaluated on 10^4 new samples from the Gaussian mixture distribution. We observe that on average the chance constraint is satisfied. A higher chance threshold, *i.e.*, allowing a higher probability of colliding with the obstacle, results in lower trajectory durations.

6.2.3 Online Planning (MPC) The online planning experiments correspond to a receding horizon/model predictive control (MPC) approach, where a new robot trajectory is planned at every MPC step $t_{i,\text{MPC}} = t_{i-1,\text{MPC}} + \Delta_{\text{MPC}}$ with

$1/\Delta_{\text{MPC}}$ being the run frequency of the MPC controller. At each MPC step, the robot gets a position update of the M obstacles in the environment, which is assumed to be exact, *i.e.*, no measurement uncertainty. As in the offline planning experiment, we assume that CC-VPSTO has access to a generative model that generates predictions of future obstacle motions, with samples that reflect the underlying uncertainty in those motions. In our online experiments we use a random walk model, parametrised to match the simulation environment. In a real-world setting, this could be replaced by a generative model learned from real-world data, *e.g.*, a model similar to Jiang et al. (2023). In our optimisation scheme, given a position update, new obstacle trajectories are sampled from the random walk model and rolled out for a fixed time horizon $T > \Delta_{\text{MPC}}$. As described in Sec. 6.1, one sample of the uncertainty in this experiment corresponds to one possible future of how the obstacles are going to evolve in the next time steps, *i.e.*, one sample maps to M trajectory predictions of duration T for M obstacles.

We evaluate online CC-VPSTO on four metrics across three environment configurations with four and five obstacles. Each obstacle is initialised with varying start position, velocity, and acceleration variance in the random walk model. The trajectory we report results on is the trajectory that the robot executes, *i.e.*, the concatenation of the first Δ_{MPC} time steps of each of the solutions across MPC steps. One single experiment produces one trajectory from the initial position to the goal for the given environment instantiation and η-value. Fig. 8 shows a qualitative example of each of the three environment configurations used in the MPC experiments, along with the respective CC-VPSTO solutions for different values of η. Note, that the length of the plotted obstacle trajectories was not fixed across the three examples, but depended on the maximum duration of the generated solutions for the given example. All solutions depicted are collision-free. Additional details about the environment configurations can be found in the Appendix in Sec. 12.1. For each environment configuration and for different values of η (0.05, 0.2, 0.4, 0.6, 0.8), we run 1000 experiments. The evaluation metrics for the experiments are:

1. ***Motion duration (time until goal reached):*** This translates to the number of MPC steps needed until the robot reaches the goal. In the experiments, we set a maximum number of 100 MPC steps. We only report the duration for successful experiments.

2. ***Success rate:*** The fraction of experiments, where the generated trajectory reaches the goal within the maximum number of MPC steps. An experiment is further only considered to be successful if the executed robot trajectory does not collide at any point of time with any of the obstacles *and* if the goal is reached.

3. ***Collision rate:*** This rate reflects the share of experiments where the respective trajectories were in collision at least once with any of the obstacles across the entire motion.

4. ***Minimum distance to obstacles:*** Per experiment, we measure the closest distance of the robot to any of the obstacles across the entire motion. This metric is only reported for successful experiments.

Note, that these metrics are different from the metrics used in the offline evaluation. This is because we aim to show the properties of the MPC trajectory given the guarantees from the offline experiments (which corresponds to a single MPC step in the online case).

Baselines We compare the use of our confidence-based bound k_{binom} as a surrogate for the chance constraint (cf. Eq. (5) and Eq. (9)) against two alternative approximations: *i)* the naïve MC approximation of the original chance constraint, as proposed in *e.g.*, Blackmore et al. (2010), and *ii)* the CVaR-based formulation described in Sec. 4.5, following the approach of Yin et al. (2023). As the main contribution of this work lies in the derivation a new confidence-bounded sample average approximation of a chance constraint, we do not compare against other methods of solving the resulting optimisation problem, such as Model Predictive Path Integral Control (MPPI) (Williams et al. 2017). For a more thorough comparison of VP-STO to MPPI, please refer to our previous work (Jankowski et al. 2023). In addition, we compare our approach to a baseline, that we abbreviate with "ML-VPSTO", where ML stands for maximum likelihood. Instead of computing the probability of constraint violation based on samples, ML-VPSTO uses the same samples to compute mean obstacle trajectories and uses standard VP-STO to generate a solution that avoids these trajectories. In addition, running CC-VPSTO with $\eta = 0$ can also be seen as a baseline, as this is comparable to using a hard collision avoidance constraint within VP-STO. For all MPC simulation experiments, we assumed a replanning frequency of 4 Hz with a time step of 0.05 seconds, while setting the planning horizon T_{MPC} to 5 seconds (mapping to 100 time steps for the rollouts), the maximum number of MPC iterations to 100 and the number of samples N to 100.

Results The results of the online experiment are a key insight of this paper, as they demonstrate the effects of combining *reactivity* (the MPC setting) with *probabilistic bounds on constraint satisfaction* (the chance constraints). Fig. 9 summarises the results across the 1000 experiments for each of the three different environment configurations. We observe that CC-VPSTO in an MPC loop is able to generate trajectories that are entirely collision-free for η values of up to 5%. In the given experimental setting, ML-VPSTO is approximately equivalent to permitting collisions with a probability as high as 80% in CC-VPSTO. In `env0`, *i.e.*, the most challenging environment configuration, both ML-VPSTO and CC-VPSTO with $\eta = 0.8$ lead to a situation where 50% of the experiments are in collision. This indicates that employing average obstacle prediction for collision avoidance is inadequate, as a 50% collision rate is generally not an acceptable outcome in the majority of robotic applications. Moreover, the dependency of the constraint satisfaction/performance trade-off on the value of η is reflected in the motion duration. The higher the value of η, the shorter the duration of the trajectory. While ML-VPSTO produces the quickest trajectories, it is also the least safe approach. For reference, we also include T_{straight} in the duration plots, which is the duration of the straight-line trajectory from start to goal (ignoring obstacles). When looking at the minimum distance to obstacles across trajectories and experiments, the expressiveness of this

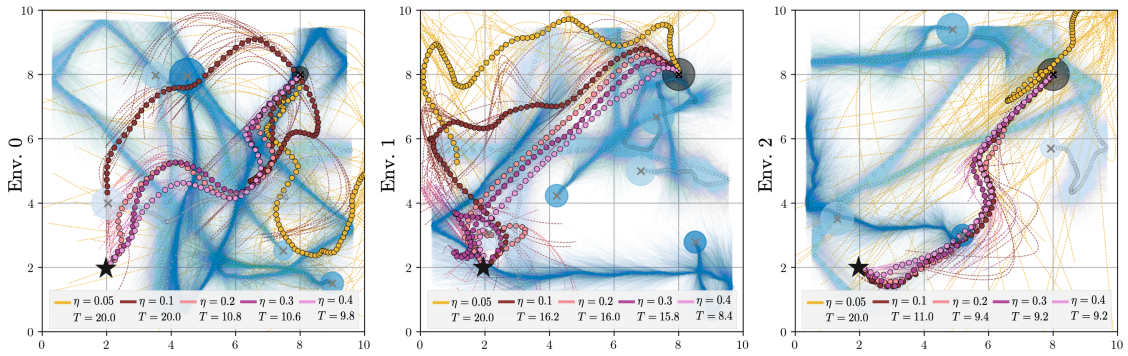


Figure 8. ***Overview of the environments used in the MPC simulation experiments.*** Each plot shows one example experiment setting for the respective environment configuration, along with the CC-VPSTO MPC solutions for varying η values from a single experiment run. The initial obstacle positions and their radii are shown as blue circles. Smaller circles along the robot trajectory mark ground truth MPC updates, with the corresponding predicted sample rollouts visualized as semi-transparent trajectories originating from each update point. The robot's start and goal are indicated by a dark grey circle (representing the robot radius) and a star, respectively. The current solution at each MPC step, *i.e.*, the trajectory segment planned over a receding horizon from the current robot position, is shown as a dashed line. Solutions get less conservative and more timing-effecient with growing values of η.

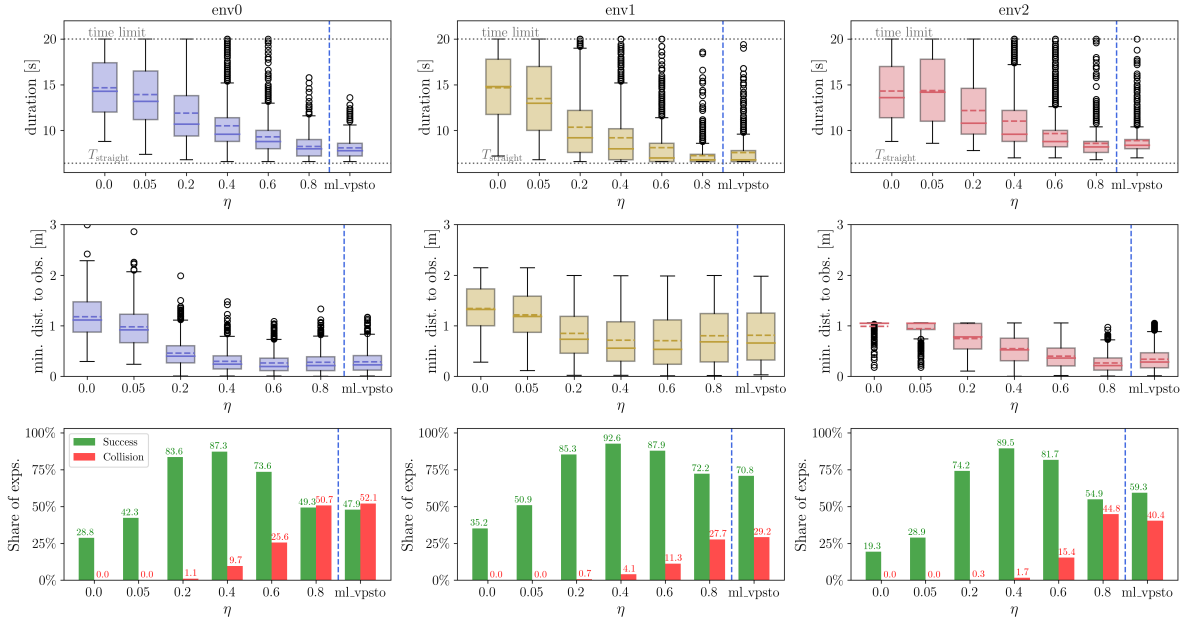


Figure 9. ***Simulation: MPC experiments.*** Evaluating *motion duration*, *success rate*, *collision rate* and the *minimum distance to obstacles* across 1000 experiments on 3 different environments. One experiment corresponds to running online-CC-VPSTO until reaching the goal or until a maximum number of 100 MPC steps is reached. Goal and start location remain fixed across all experiments and environments, whilst the obstacle trajectories vary across experiments and environments. Each environment is initialized with different start positions and velocities of the obstacles, as well as different variance on the acceleration used in the random walk model. The boxplots include the mean (dashed line) and median (solid line) across all experiments.

metric depends on the environment configuration. For the first and second environment configuration, there is a decreasing trend, until it plateaus at values of $\eta > 0.4$ for the first environment. For the second environment configuration, after a downward trend, the minimum distance to obstacles increases again for values of $\eta > 0.4$. This can be explained by CC-VPSTO being more risk-taking and probably choosing a more direct path to the goal, which can possibly lead to more collisions, but in the case of no collision, the distance to obstacles might actually be bigger, as the motion is also quicker and some obstacles might not have had time to move closer to the robot. Last, the small variance in the distances for small η values in the third environment can be explained by the initial configuration of obstacles, as they are already very close to the robot, which is then probably already the minimum distance across the entire motion. Overall, for this experiment setting, it seems like CC-VPSTO with $\eta = 0.4$ offers a good trade-off between constraint satisfaction and performance, as it is able to generate trajectories with a high success rate, whilst also being able to generate trajectories that are efficient in their motion duration.

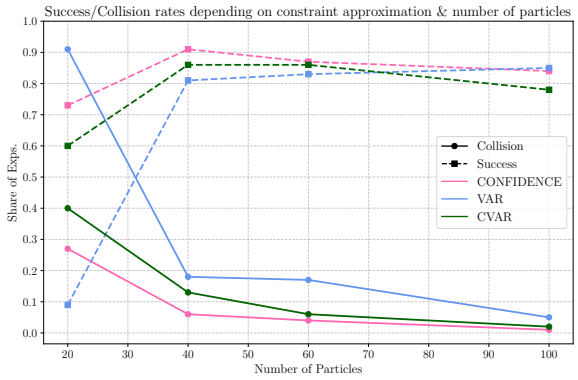


Figure 10. Comparison of the confidence-bounded chance constraint approximation to CVaR and a standard VaR approximation in terms of success and collision rates across MPC experiments depending on the number of samples used in the MC approximation.

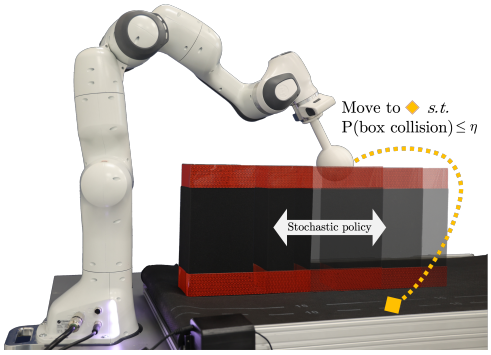


Figure 11. ***Robot experiment setup.*** The robot task is to move from one side to the other side of the conveyor belt while assuring that the probability of colliding with the box obstacle is below a user-defined threshold η. The motion of the box obstacle is stochastic, as the conveyor belt is actuated with constant velocity, but the rate of direction change follows an exponential distribution.

Last, we evaluate the effect of the number of samples used in the MC approximation on the success and collision rates depending on the surrogate constraint that is used in CC-VPSTO. We do this across 1000 MPC experiments for each sample count and environment configuration. We evaluate three different surrogate constraints:

1. *Confidence:* $s_N(\boldsymbol{x}, D) \leq \eta_{\text{binom}}(\eta, \beta)$
2. *Value at Risk (VaR):* $s_N(\boldsymbol{x}, D) \leq \eta$
3. *Conditional VaR (CVaR):* $s_N(\boldsymbol{x}, D) \leq \text{CVaR}_{\text{max}}$, with CVaR_{max} set to 0.6.

For all experiments we fixed η to 0.2 and the confidence threshold to 0.99. The results are shown in Fig. 10. The numbers are averaged across three different environment configurations. For 100 samples, the experiments are equivalent to the results shown in Fig. 9. With more samples, the collision rate converges to zero due to the MPC setting. The results show that the confidence-bounded approximation is the only approximation that is able to maintain high success and low collision rates with a small number of samples. This is in contrast to the other methods which do not account for the number of samples used in the approximation. With an increasing number of samples, VaR and our approximation converge to the same success and collision rates, which is expected as the number of samples increases, as shown in Sec. 4.5. Moreover, we notice that the success rate of CVaR decreases as the number of samples increases. Since CC-VPSTO explicitly accounts for the number of samples used in the approximation, its performance is less sensitive to the choice of N, showing the lowest variation across sample counts among the evaluated methods. That said, in low-sample regimes, some approximation error remains unavoidable.

6.3 Robot Experiment

We further demonstrate CC-VPSTO on a real robot for the scenario shown in Fig. 11. The robot is tasked to move from one side to the other of the conveyor belt, whilst avoiding a box which is controlled according to a stochastic policy. This requires online, reactive motion generation that balances constraint satisfaction with task efficiency. The possible motions, also illustrated in Fig. 1, are to either move behind or in front of the box, as the robot is not allowed to simply move over the box. Moreover, besides the candidate trajectories that we synthesise from the sampled via-points in CC-VPSTO, we add a "waiting" trajectory to the set of candidate trajectories sampled in the final optimisation step. A waiting trajectory is a trajectory repeating the current robot position for the entire planning horizon, *i.e.*, keeping the robot stationary. This is to allow the robot to wait for the box to pass, which is a safe but not very efficient solution. Without these waiting trajectories, CC-VPSTO would keep the robot moving at all times, but this is not always necessary.

Setup. The experiment is performed on a Franka Emika robot arm. The framework was run on Ubuntu 20.04 with an Intel Core i7-8700 CPU@3.2GHz and 16GB of RAM. The ground truth box position is tracked using an Intel RealSense camera and a barcode detection pipeline. In every MPC step the robot is given the current position of the box and then plans a new trajectory using CC-VPSTO[¶]. With this setup, we are able to run the framework at a frequency of 3 Hz, using $N = 100$ samples and a planning horizon of $T_{\text{MPC}} = 3$ seconds (mapping to 60 time steps for the rollouts, as we use a time step of 0.05 seconds).

Stochastic conveyor belt policy. In this experiment, the uncertainty stems from the movement of the box on the conveyor belt, which serves as an obstacle for a robot to navigate around. The conveyor belt is velocity controlled, where the magnitude of the velocity is fixed to $0.05\frac{\text{m}}{\text{sec}}$ but its direction is governed by the probability density function $f(x; \alpha) = \alpha \exp(-\alpha x)$, where x is the time since the last direction change and α is a parameter influencing the rate of direction change. We describe our implementation of this stochastic model in more detail in the appendix in Sec. 12.3.

[¶]Note we ignore measurement noise in this setup.

Results Similar to the MPC experiments in simulation, we evaluate our real-world robot experiment on *i)* the motion duration per run, *i.e.*, the time taken to go from one side of the conveyor belt to the other side, *ii)* the share of experiments that collide with the box, and *iii)* the minimum distance to the box across 70 runs for different values of η, as shown in Fig. 12. We do not compare our approach to "ML-VPSTO" as for the given stochastic model, the mean over the exponential distribution is not useful. However, we include $\eta = 0$ as a baseline, which corresponds to a VPSTO approach with a hard collision avoidance constraint. Overall, the results support the insights gained from the simulation experiments. We observe that the higher the value of η, the shorter the duration of the trajectory. This is because higher values of η allow CC-VPSTO to generate trajectories that are more efficient, but also less safe. Moreover, we also observe the trend that higher values of η indeed lead to a higher share of experiments that collide with the box. However, we also observe that the share of experiments that collide with the box is still very low, even for very high values of η. This is an interesting insight from combining MPC with chance-constrained trajectory optimisation. In addition, we observe in this experiment that a value of $\eta = 0.2$ outperforms $\eta = 0.1$ in terms of the share of experiments that collide with the box. We anticipate that additional experiments will reduce this variability, as the current variance in the results is still quite high. Last, in terms of the minimum distance to the box, we cannot observe a clear trend across different values of η. This can be explained by the use of waiting trajectories that do not move the robot at all. The robot can choose to wait very close to the box, which is a constraint-satisfying but not very efficient solution. This results in higher durations, but not higher minimum distances.

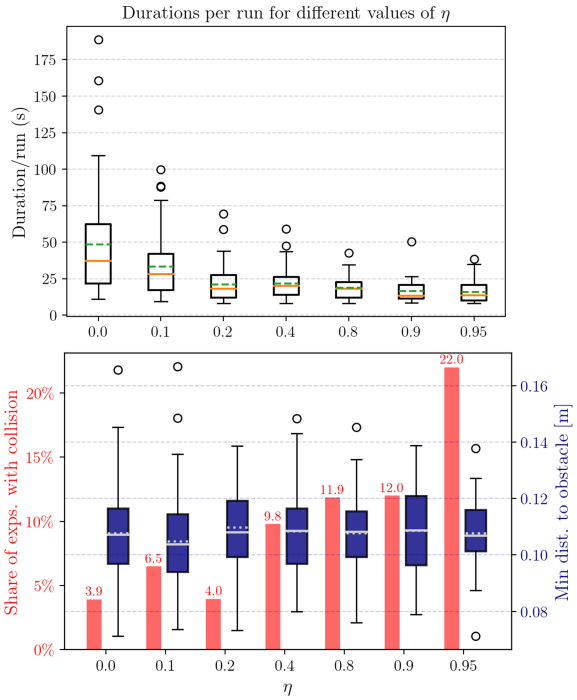


Figure 12. ***Robot experiment results.*** Similar to the metrics evaluated in the MPC simulation experiments, we evaluate the *motion duration*, *minimum distance* to the box and the *share of experiments that collide* with the box across 70 experiments for different values of η. Means in the boxplots are shown as dashed lines and medians as solid lines.

7 Discussion and Future Work

Our experiments show that CC-VPSTO is able to generate task-efficient motions while consistently bounding the probability of constraint violation, *e.g.*, collision with stochastic obstacles. While it is typically more challenging to deal with chance constraints over entire trajectories (as opposed to constraints per time step), our SAA formulation allows us to do this in a straightforward way by simulating trajectories of the obstacles and the robot then checking for collisions between them at any time over a given horizon. This is made possible by the flexibility of our approach which makes no assumption on the distribution of the uncertainty, but only requires sampling access to it. Hence, this can also be a joint distribution across all sources of uncertainty, *e.g.*, several obstacles.

Consistent with our theoretical insights, the collision rate in the experiments remained below the specified threshold η, despite the fact that the independence assumption does not strictly hold in the receding-horizon (MPC) setting. Nevertheless, we observe a gap between the collision rate and the threshold η. This gap is much bigger in the online planning (MPC) than in the offline planning experiments. This can be explained by the fact that at each MPC step we optimise trajectories for a longer horizon than just the time steps that we actually execute on the robot. Hence, the anticipation of potential collisions in the future makes us more conservative, resulting in lower actual rate of collision than that which was imposed. In future work, we plan to use discounted probabilities in the chance constraint, such as in Yan et al. (2018), to allow for larger probabilities of collision for time steps far in the future, knowing that the control input that we will apply then will be recomputed with stronger constraints in the meantime. Besides the introduction of discounted chance constraints, another direction could be to explore how to adapt the parameters η and β during online execution, *e.g.*, based on the current state of the system, the current uncertainty distribution, or the current cost function. This could result in a more robust and adaptive approach, which would be more suitable for real-world applications.

An important direction for future research is how to respond when a constraint violation actually occurs, especially since our framework allows such violations with some probability. While we do not propose an explicit or general mechanism for handling these situations, we assume that the robot is capable of recovering from violations. In this context, the MPC framework provides implicit robustness through continual replanning based on updated information about the uncertain environment. Nonetheless, it remains an open and valuable question how the planner's objective might be adapted in the face of violations; for instance, by temporarily shifting the objective to minimize the probability of further constraint violations rather than pursuing the original task goal.

Moreover, our work does not ensure recursive feasibility, which enforces that there always exists a solution to the

optimisation problem. Other works such as Köhler et al. (2023), have addressed this issue by enforcing that the predicted nominal state lands in a terminal set while not taking the measured state into account. However, this again comes at the cost of being restricted to linear systems with linear inequality constraints and convex objectives. For chance constraints this means that they would need to be linearized into half-space constraints, yet if we took the collision avoidance examples from this work and view them as a system that is augmented by the obstacle states, the non-collision constraint itself is non-linear in the augmented state as it has a quadratic relation through the distance measure. Yet, an interesting direction for future research would be how to ensure recursive feasibility through a less constrained problem formulation.

A core assumption in our approach is that we are given a *representative* model of the uncertainty, from which we can take samples. Yet, this might be a limitation in practice, as our model might not capture the true underlying distribution with sufficient accuracy, relating to *epistemic uncertainty*. However, that can be addressed in future work, by either extending the approach to be *distributionally robust*, such as in Hakobyan and Yang (2021), or by using generative data-driven models that can be adapted online as the robot acquires more data, similar to the work of Thorpe et al. (2022). In addition, future work should also quantify the extent to which MPC is able to provide inherent robustness, given that its closed-loop formulation allows for partial compensation when the true uncertainty distribution diverges moderately from the one used during optimisation.

In terms of computational efficiency, we have demonstrated the applicability of our algorithm to an MPC setting, where we achieve frequencies of 3 Hz on a real robot using 100 samples. The biggest computational bottleneck lies in the rollouts of the uncertainty dynamics. We therefore believe that the reported control rate can be improved by adding parallelization and GPU acceleration, which we did not leverage in the given experiments. However, multimodal distributions require more samples in the approximation, which might further limit current control rates. We believe that future work could explore methods to efficiently generate representative sample sets, possibly leveraging learned generative models. The advantage of our formulation is that the user can actively choose the number of samples while considering computational resources and requirements of minimum control rates. We also believe that there is still room for improvement in our implementation, as the sample rollouts for the stochastic box model have not been parallelised, as was done in the simulation experiments. Last, a learned generative model could also further improve the computational efficiency of the rollouts over the current Monte Carlo simulations, depending on the model's inference speed.

8 Conclusion

In this work, we addressed the problem of robot motion planning under uncertainty, aiming for both efficiency and constraint satisfaction in stochastic control settings. We introduced a novel surrogate formulation for chance-constrained optimisation that enables statistically sound sampling-based motion planning under uncertainty. This, in turn, supports integration into a Model Predictive Control (MPC) framework for *online*, reactive robot control. The strength of our approach lies in its generality, as it does not require any specific assumptions on the underlying uncertainty distribution, the dynamics of the system, the cost function or the specific form of inequality constraints. While we focused on the problem of collision avoidance in this work, our approach is not limited to this problem, as it can be applied to any type of stochastic control problem, as long as we can sample from the uncertainty distribution. For instance, in future work we aim to extend this framework to include constraints on interaction forces in the context of contact-rich manipulation tasks and physical human-robot interaction. We showed that our approach is able to generate efficient trajectories that satisfy probabilistic constraints with high confidence across a variety of scenarios, including a real-world robot experiment.

References

Abdi H et al. (2007) Bonferroni and šidák corrections for multiple comparisons. *Encyclopedia of measurement and statistics* 3(01): 2007.

Alcan G and Kyrki V (2022) Differential dynamic programming with nonlinear safety constraints under system uncertainties. *IEEE Robotics and Automation Letters* 7(2): 1760–1767.

Badings T, Romao L, Abate A, Parker D, Poonawala HA, Stoelinga M and Jansen N (2023) Robust control for dynamical systems with non-gaussian noise via formal abstractions. *Journal of Artificial Intelligence Research* 76: 341–391.

Berger GO, Jungers RM and Wang Z (2021) Chance-constrained quasi-convex optimization with application to data-driven switched systems control. In: *Learning for Dynamics and Control*. PMLR, pp. 571–583.

Bhardwaj M, Sundaralingam B, Mousavian A, Ratliff ND, Fox D, Ramos F and Boots B (2022) Storm: An integrated framework for fast joint-space model-predictive control for reactive manipulation. In: *Conference on Robot Learning*. PMLR, pp. 750–759.

Blackmore L (2006) A probabilistic particle control approach to optimal, robust predictive control. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. p. 6240.

Blackmore L, Li H and Williams B (2006) A probabilistic approach to optimal robust path planning with obstacles. In: *2006 American Control Conference*. IEEE, pp. 7–pp.

Blackmore L, Ono M, Bektassov A and Williams BC (2010) A probabilistic particle-control approximation of chance-constrained stochastic predictive control. *IEEE Transactions on Robotics* 26(3): 502–517.

Calafiore GC (2010) Random convex programs. *SIAM Journal on Optimization* 20(6): 3427–3464.

Calafiore GC and Campi MC (2006) The scenario approach to robust control design. *IEEE Transactions on automatic control* 51(5): 742–753.

Campi MC and Garatti S (2011) A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality. *Journal of optimization theory and applications* 148(2): 257–280.

Castillo-Lopez M, Ludivig P, Sajadi-Alamdari SA, Sanchez-Lopez JL, Olivares-Mendez MA and Voos H (2020) A real-time approach for chance-constrained motion planning with dynamic obstacles. *IEEE Robotics and Automation Letters* 5(2): 3620–3625.

Dai S, Schaffert S, Jasour A, Hofmann A and Williams B (2019) Chance constrained motion planning for high-dimensional robots. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 8805–8811.

de Groot O, Ferranti L, Gavrila D and Alonso-Mora J (2023) Scenario-based motion planning with bounded probability of collision. *arXiv preprint arXiv:2307.01070* .

Hakobyan A and Yang I (2021) Wasserstein distributionally robust motion control for collision avoidance using conditional value-at-risk. *IEEE Transactions on Robotics* 38(2): 939–957.

Hansen N (2016) The CMA evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772* .

Heirung TAN, Paulson JA, O'Leary J and Mesbah A (2018) Stochastic model predictive control—how does it work? *Computers & Chemical Engineering* 114: 158–170.

Homem-de Mello T and Bayraksan G (2014) Monte carlo sampling-based methods for stochastic optimization. *Surveys in Operations Research and Management Science* 19(1): 56–85.

Jankowski J, Brudermüller L, Hawes N and Calinon S (2023) VP-STO: Via-point-based stochastic trajectory optimization for reactive robot behavior. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 10125–10131.

Jankowski J, Racca M and Calinon S (2022) From Key Positions to Optimal Basis Functions for Probabilistic Adaptive Control. *IEEE Robotics and Automation Letters* 7(2): 3242–3249.

Janson L, Schmerling E and Pavone M (2017) Monte carlo motion planning for robot trajectory optimization under uncertainty. In: *Robotics Research: Volume 2*. Springer, pp. 343–361.

Jasour AM, Aybat NS and Lagoa CM (2015) Semidefinite programming for chance constrained optimization over semialgebraic sets. *SIAM Journal on Optimization* 25(3): 1411–1440.

Jiang C, Cornman A, Park C, Sapp B, Zhou Y, Anguelov D et al. (2023) Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 9644–9653.

Köhler J, Geuss F and Zeilinger MN (2023) On stochastic mpc formulations with closed-loop guarantees: Analysis and a unifying framework. In: *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, pp. 6692–6699.

Lew T, Bonalli R and Pavone M (2023) Risk-averse trajectory optimization via sample average approximation. *IEEE Robotics and Automation Letters* .

Majumdar A and Pavone M (2020) How should a robot assess risk? towards an axiomatic theory of risk in robotics. In: *Robotics Research: The 18th International Symposium ISRR*. Springer, pp. 75–84.

Majumdar A and Tedrake R (2017) Funnel libraries for real-time robust feedback motion planning. *The International Journal of Robotics Research* 36(8): 947–982.

Margellos K, Goulart P and Lygeros J (2014) On the road between robust optimization and the scenario approach for chance constrained optimization problems. *IEEE Transactions on Automatic Control* 59(8): 2258–2263.

Mesbah A (2016) Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine* 36(6): 30–44.

Mohri M, Rostamizadeh A and Talwalkar A (2018) *Foundations of machine learning*. MIT press.

Nemirovski A and Shapiro A (2007) Convex approximations of chance constrained programs. *SIAM Journal on Optimization* 17(4): 969–996.

Ono M and Williams BC (2008) Iterative risk allocation: A new approach to robust model predictive control with a joint chance constraint. In: *2008 47th IEEE Conference on Decision and Control*. IEEE, pp. 3427–3432.

Pagnoncelli BK, Ahmed S and Shapiro A (2009) Sample average approximation method for chance constrained programming: theory and applications. *Journal of optimization theory and applications* 142(2): 399–416.

Parsi A, Anagnostaras P, Iannelli A and Smith RS (2022) Computationally efficient robust mpc using optimized constraint tightening. In: *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, pp. 1770–1775.

Peña-Ordieres A, Luedtke JR and Wächter A (2020) Solving chance-constrained problems via a smooth sample-based nonlinear approximation. *SIAM Journal on Optimization* 30(3): 2221–2250.

Prékopa A (2013) *Stochastic programming*, volume 324. Springer Science & Business Media.

Priore S and Oishi M (2023) Chance constrained stochastic optimal control based on sample statistics with almost surely probabilistic guarantees. *arXiv preprint arXiv:2303.16981* .

Schildbach G, Fagiano L, Frei C and Morari M (2014) The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations. *Automatica* 50(12): 3009–3018.

Schmerling E and Pavone M (2016) Evaluating trajectory collision probability through adaptive importance sampling for safe motion planning. *arXiv preprint arXiv:1609.05399* .

Shalev-Shwartz S and Ben-David S (2014) *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shapiro A (2003) Monte carlo sampling approach to stochastic programming. In: *ESAIM: proceedings*, volume 13. EDP Sciences, pp. 65–73.

Shapiro A, Dentcheva D and Ruszczynski A (2021) *Lectures on stochastic programming: modeling and theory*. SIAM.

Sun W, Torres LG, Van Den Berg J and Alterovitz R (2016) Safe motion planning for imprecise robotic manipulators by minimizing probability of collision. In: *Robotics Research: The 16th International Symposium ISRR*. Springer, pp. 685–701.

Taboga M (2017) Lectures on probability theory and mathematical statistics. *(No Title)* .

Thorpe A, Lew T, Oishi M and Pavone M (2022) Data-driven chance constrained control using kernel distribution embeddings. In: *Learning for Dynamics and Control Conference*. PMLR, pp. 790–802.

Trevisan E, Mustafa KA, Notten G, Wang X and Alonso-Mora J (2025) Dynamic risk-aware mppi for mobile robots in crowds via efficient monte carlo approximations. *arXiv preprint*

arXiv:2506.21205 .

Van Den Berg J, Abbeel P and Goldberg K (2011) Lqg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information. *The International Journal of Robotics Research* 30(7): 895–913.

Vincent JA, Feldman AO and Schwager M (2024) Guarantees on robot system performance using stochastic simulation rollouts. *IEEE Transactions on Robotics* .

Wang A, Jasour A and Williams B (2020) Moment state dynamical systems for nonlinear chance-constrained motion planning. *arXiv preprint arXiv:2003.10379* .

Williams G, Wagener N, Goldfain B, Drews P, Rehg JM, Boots B and Theodorou EA (2017) Information theoretic mpc for model-based reinforcement learning. In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 1714–1721.

Yan S, Goulart P and Cannon M (2018) Stochastic model predictive control with discounted probabilistic constraints. In: *2018 European Control Conference (ECC)*. IEEE, pp. 1003–1008.

Yin J, Zhang Z and Tsiotras P (2023) Risk-aware model predictive path integral control using conditional value-at-risk. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 7937–7943.

Zhang Z, Tomlinson J and Martin C (1997) Splines and linear control theory. *Acta Math. Appl* 49: 1–34.

5.1 Supplementary Discussion

Bayesian Interpretation Our confidence-bounded surrogate constraint in Sec. 4 is fundamentally frequentist: we treat each constraint evaluation as a Bernoulli trial and derive a binomial *confidence bound* that guarantees

$$s_N(\boldsymbol{x}; D) \leq k_{\text{binom}}(\beta, N, \eta) \tag{5.1}$$

with confidence $1 - \beta$. This yields a distribution-free guarantee that depends only on the sampling process and avoids any prior assumptions.

A related but conceptually distinct perspective arises from *Bayesian inference*(Gelman et al., 1995). In a Bayesian formulation, the (unknown) violation probability

$$p \;=\; \mathbb{P}_{\delta\sim\Delta}\Big[g(\boldsymbol{x}, \boldsymbol{\delta}) > 0\Big] \tag{5.2}$$

is itself treated as a random variable. Placing a Beta prior $\text{Beta}(\alpha, \beta)$ over p, observation of s violations in N i.i.d. samples yields the Beta–Binomial posterior

$$p \mid s \;\sim\; \text{Beta}(\alpha + s,\; \beta + N - s). \tag{5.3}$$

The Bayesian analogue of the chance constraint $p \leq \eta$ would require the posterior *credible interval* (Brown et al., 2001) to lie below the risk threshold, i.e.,

$$\Pr(p \leq \eta \mid s) \;\geq\; 1 - \beta. \tag{5.4}$$

The structure of this Bayesian update closely mirrors the frequentist construction: inverting the Beta posterior cumulative distribution function (CDF) under a uniform prior $\text{Beta}(1, 1)$ produces a bound numerically close to the classical Clopper–Pearson *confidence* interval, which itself is obtained by inverting the binomial CDF. This parallel highlights that both approaches lead to bounds of similar mathematical form. However, whereas the Bayesian credible interval depends on the choice of prior, our method uses the binomial model directly and therefore avoids prior specification while providing distribution-free confidence guarantees. In summary,

while a Bayesian credible-interval construction offers an alternative interpretation of uncertainty in the violation probability, our approach deliberately adopts the frequentist viewpoint: we use binomial confidence bounds to quantify sampling error without introducing prior assumptions. Yet, it would be an interesting direction for future work to explore the incorporation of prior knowledge about constraint satisfaction into the chance-constrained MPC framework through a Bayesian approach.

Relation to Probabilistic Control Barrier Functions Control Barrier Functions (CBFs) offer an alternative approach to safety-critical control by enforcing forward-invariance of a safe set through Lyapunov-style constraints (Ames et al., 2016). While classical CBFs assume deterministic systems, a growing body of work extends them to stochastic settings, where safety is enforced either in expectation or with probabilistic guarantees (Clark, 2021; Wang et al., 2021a). Recent work has also explored learning-based CBFs that infer barrier certificates or uncertainty models from data, e.g., (Mestres et al., 2025). While probabilistic CBFs also provide probabilistic guarantees, they typically impose structural assumptions on dynamics (e.g., control-affine forms) and enforce pointwise probabilistic safety. In contrast, our chance-constrained formulation enforces a joint trajectory-level probability bound over the entire horizon, does not rely on a control-affine or smooth constraint structure, operates with arbitrarily complex uncertainty distributions via Monte-Carlo sampling, and integrates directly with stochastic optimisation.

5.2 Limitations and Future Work

The main paper provides an extensive discussion of several limitations and future directions, including distributional robustness, computational efficiency, recursive feasibility, dynamic risk adjustment, and the integration of learning-based components. However, we have not considered two alternative ways of incorporating learning into chance-constrained MPC, which could be promising topics for future work and are briefly discussed below.

Learning Chance-Constrained Control Policies A promising direction that naturally extends from the generative predictive control framework presented in Chapter 4 is to learn chance-constrained control policies directly from offline data. CC-VPSTO could be used to generate a large dataset of chance-constrained MPC control sequences, which could then be used to train a conditional generative model that maps states directly to chance-constrained control sequences. This would enable larger sample sizes and longer planning horizons, even for tasks requiring expensive forward simulations, while allowing the learned policy to be executed in real time without the need for online optimisation.

Learning Chance Constraint Evaluation from Data An alternative to the sampling-based formulation presented in this chapter is to learn chance constraint evaluations directly from data. Instead of relying on empirical sampling to approximate the constraint satisfaction probability, one could train a model to predict feasible regions or constraint satisfaction probabilities conditioned on the system state, uncertainty, or control inputs. This data-driven approach can yield constraints that are smoother and more efficiently evaluable during planning, thereby alleviating some of the computational burdens associated with Monte Carlo sampling. For example, recent work (Moss et al., 2024) has employed neural networks as surrogate constraints in the context of planning in chance-constrained POMDPs. By learning constraint satisfaction from rollout data, this method bypasses the reliance on extensive empirical sampling within a framework like MCTS, instead using a dedicated network head to output the failure probability conditioned on the current belief state. This allows for efficient and direct evaluation of the chance constraint during online search. Extending these ideas to the setting of chance-constrained MPC could allow for learning uncertainty-aware constraint models from demonstrations or simulation data, bridging model-based and data-driven formulations. However, we also note that these benefits come with the usual caveat that learned predictors may behave unreliably under out-of-distribution conditions, potentially compromising theoretical guarantees on constraint satisfaction.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	CC-VPSTO: Chance-Constrained Via-Point-Based Stochastic Trajectory Optimisation for Online Robot Motion Planning under Uncertainty	
Publication Status	□ Published	□ Accepted for Publication
	☑ Submitted for Publication	□ Unpublished and unsubmitted work written in a manuscript style
Publication Details	Brudermüller, L., Berger, G. O., Jankowski, J., Bhattacharyya, R., Calinon, S., Jungers, R. M., and Hawes, N. (2025a). CC-VPSTO: Chance-Constrained Via-Point-Based Stochastic Trajectory Optimisation for Online Robot Motion Planning under Uncertainty. Manuscript under review at The International Journal of Robotics Research.	

Student Confirmation

Student Name:	Lara Brudermüller		
Contribution to the Paper	– I led the development of the idea in collaboration with Guillaume Berger, Julius Jankowski, Raphaël Jungers and Nick Hawes. – I implemented all algorithms and conducted all experiments in simulation and on hardware. – I led the writing in collaboration with Guillaume Berger, who provided substantial support with the mathematical proofs and the Rademacher-bound baseline. – I incorporated feedback to the paper from Guillaume Berger, Julius Jankowski, Raphaël Jungers and Nick Hawes.		
Signature		Date	September 17, 2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Professor Nick Hawes	
Supervisor comments I confirm that Lara made a substantial contribution to this publication, and that the description above is accurate.		
Signature		Date: September 17, 2025

This completed form should be included in the thesis, at the end of the relevant chapter.

6

Actively Reducing Uncertainty via Belief-Space Control through Contacts

Publication Note

This chapter presents the work from the following submission:

Brudermüller, L., Jankowski, J., Toussaint, M., and Hawes, N. (2025c). Touch-based object localisation with spatially-aware belief entropy estimation. Manuscript under review at *IEEE International Conference on Robotics and Automation (ICRA)*

Supplementary Material

The supplementary video is available at: `https://youtu.be/RJhPJi790BM`.

In the previous chapters, we have successively extended sampling-based model predictive control methods to handle increasing levels of uncertainty in the robot's interaction with the environment. While Chapter 3 focused on improving reactivity through efficient trajectory optimisation and Chapter 5 introduced explicit uncertainty handling via chance constraints, both approaches primarily aimed to ensure robust task execution in the presence of stochastic dynamics. In contrast, the work presented in this chapter represents the first instance in this thesis where uncertainty itself becomes an *explicit component of the control objective*, with the primary goal of actively reducing it through purposeful interaction.

Actively reducing uncertainty poses unique challenges, particularly in contact-

rich scenarios where actions can both decrease and increase uncertainty depending on the outcome of stochastic interactions. This dual nature of interaction makes the problem fundamentally different from the settings addressed in previous chapters, where uncertainty was treated as an external disturbance to be mitigated rather than a quantity to be optimised. To address this, we move beyond stochastic but fully observable systems and consider *partially observable* environments, where the robot has only indirect access to the true system state through noisy or incomplete sensory feedback. We formulate the problem within a *belief space control* framework, where the robot maintains a probabilistic belief over the state and plans actions that simultaneously achieve task objectives and reduce uncertainty.

	Chapter 3	**Chapter 4**	**Chapter 5**	**Chapter 6**
Theme	Reactivity	Learning for Reactivity	Robustness	Exploration
Uncertainty Handling	Implicit	Implicit	Explicit	Explicit
Prior Knowledge of Uncertainty	None	None	Sampling-based access	Sampling-based access
Environment	Deterministic	Deterministic	Stochastic	Stochastic
Method	MPC	MPC	MPC	Belief-space control
Control Problem	Min. cost	Min. cost	Min. cost s.t. chance constraints	Max. information gain

Table 1.2: Theme, settings and methods for each chapter.

Specifically, this chapter presents a framework for touch-based object localisation and manipulation under partial observability, in which uncertainty reduction is guided by an information-theoretic objective. By incorporating a non-parametric differential entropy estimator directly into the control loop, the robot can evaluate and select actions that are expected to yield the greatest reduction in uncertainty over the object's pose. This represents a shift from reactive or robust control to *explicit* uncertainty handling through information-seeking behaviour, where *exploration* is an integral part of achieving reliable manipulation in uncertain environments.

The belief representation used in this work is non-parametric, and the proposed entropy estimator operates directly on its sampled particles. As a result, we require only *sampling-based access* to the underlying uncertainty distributions, without assuming any specific parametric form or known moments, representing the least restrictive setting and supporting the thesis goal of developing broadly generalizable approaches. We summarise this setting in Table 1.2, which is repeated above.

We demonstrate the proposed approach in simulated and real-world experiments, showing that the robot can effectively reduce uncertainty through solely proprioceptive feedback from contact interactions, enabling accurate object localisation and manipulation without relying on vision or tactile sensors. The exclusive use of proprioceptive feedback in this work is motivated by an interest in exploring the limits of what can be achieved using only proprioceptive sensing, without relying on external modalities such as vision or tactile sensors. While many existing approaches to contact-rich manipulation assume access to high-resolution touch or visual feedback, these sensors can be expensive, fragile, or unavailable in unstructured environments. In contrast, proprioception, e.g., joint angles, velocities, and torques, is universally available on robotic platforms and provides a robust and low-latency signal. By developing methods that infer contact events and reduce uncertainty using proprioception alone, we aim to broaden the applicability of autonomous manipulation in settings where external sensing is unreliable or infeasible. This minimalist sensing setup presents a significant challenge, as contact must be inferred indirectly from subtle changes in the robot's internal state. However, it also offers a unique opportunity to build highly generalisable and robust control strategies that make the most of the information already inherent in the robot's motion and interaction dynamics. Additional implementation details are provided in the Appendix of the paper manuscript, which is included at the end of this thesis as Appendix C.

Touch-Based Object Localisation with Spatially-Aware Belief Entropy Estimation

Lara Brudermüller[1], Julius Jankowski[2], Marc Toussaint[3], and Nick Hawes[1]

***Abstract*— Robust robotic manipulation in the real world requires coping with incomplete or unreliable sensory input. While vision provides rich information, it often fails in the presence of occlusions, clutter, or poor lighting. In such cases, touch offers a robust alternative, enabling object localisation through contact alone. We present a touch-only global localisation method that operates in continuous state space with a particle belief. Sparse contact/no-contact signals are turned into informative likelihoods via a proximity-aware measurement model, and contact-aware resampling mitigates particle starvation. An information-gathering controller selects actions that maximise expected information gain using a non-parametric entropy estimator sensitive to both observation updates and dynamics. On real hardware, the system reliably localises and then grasps from broad, multi-modal initial beliefs with mode separations up to 0.4 m, far beyond the narrow uncertainty ranges assumed in related work. Information-aware localisation-actions speed up belief convergence and boost grasp success; and ablations in simulation confirm the benefits of the measurement and resampling components.**

I. Introduction

Humans can manipulate objects using only touch, even in the absence of vision. For robots to approach similar capabilities in unstructured or visually-challenging environments, such as those involving occlusions, clutter, or poor lighting, localising objects through touch alone represents a promising alternative. This has driven the development of algorithms that refine an object's pose estimate through deliberate physical interaction [1]–[3]. As an illustrative example, consider a robot having to retrieve a keyring from inside a bag using a multi-fingered hand. The object pose is initially unknown, contacts are intermittent and ambiguous, and the object may move such that exploratory interactions can either resolve ambiguity or further increase uncertainty in the object state. Such examples highlight the challenges of high-dimensional contact-rich estimation problems, where the robot often starts with little information. Estimating the posterior over object pose in these settings is computationally demanding: the complexity grows rapidly with both the number of degrees of freedom (DOFs) and the size of the initial uncertainty region [4]. As a result, most prior approaches restrict either the problem dimensionality or the scale of initial uncertainty. Yet, contact-rich manipulation is not only characterized by high-dimensional state and action spaces, but also the inherent highly non-linear contact dynamics and the multi-modality of the system state distributions [5]. Discretising the problem space addresses the multi-modality while enabling standard filtering and planning over finite hypotheses [3], [6], but the curse of dimensionality limits the respective resolution and thus fails to capture the rich contact dynamics present in such manipulation tasks. Instead, these tasks require a framework that plans *in continuous state space*, anticipates *what* will be sensed and *how* actions reshape uncertainty through interaction, and operates from *uninformed, non-parametric* priors with broad support.

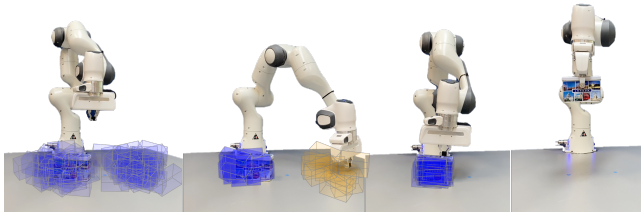


Fig. 1. Experimental setup of blind grasping. *Left to right:* initial particle belief with uniform weights (■); information-gathering trajectory rejecting particle hypotheses (■); converged belief after contact; successful grasp.

Towards this end, we address touch-based localisation with a system that operates in continuous state spaces and actively refines the belief distribution through contact. We propose a particle filter with a proximity-aware measurement model to turn sparse binary proprioceptive contact signals into informative likelihoods combined with contact-aware resampling. An information-gathering controller predicts how actions reshape the belief and selects trajectories that maximise expected information gain using a non-parametric entropy estimator that not only considers the probabilities of different object poses but also their spatial density.

Contributions: We address continuous-space object localisation through contact from uninformed, non-parametric beliefs, making the following contributions:

1) A *touch-only global localisation system* that plans and estimates directly in continuous state space with beliefs from uninformed, non-parametric priors, suitable for operation when vision is unreliable or absent.
2) A *proximity-aware measurement model* for contact that converts sparse binary signals into informative likelihoods, and a *contact-aware resampling* strategy that mitigates particle starvation under discontinuous observations.
3) A *sampling-based information-gathering controller* that selects candidate probing actions based on a non-parametric *differential entropy estimator* that captures both observation-driven changes (weights) and dynamics-driven changes (spatial density) in the belief.

[1]Oxford Robotics Institute, University of Oxford, UK; {larab, nickh}@robots.ox.ac.uk
[2]Idiap Research Institute & Ecole Polytechnique Fédérale de Lausanne (EPFL), CH; jankowskijulius@gmail.com
[3]TU Berlin, Germany; toussaint@tu-berlin.de

On real hardware (cf. setup in Fig. 1) and in simulation, the approach localises and grasps reliably under broad, *multi-modal* initial non-parametric beliefs with separations up to $0.4\,\mathrm{m}$, far beyond the narrow uncertainty ranges (typically up to $0.04\,\mathrm{m}$) assumed in related work, highlighting the significant benefit of continuous touch-based object localisation when vision is unreliable or unavailable.

II. Related Work

a) Object Localisation through Contact: Early work in robot manipulation primarily relied on visual feedback, but as we move towards more robust and dexterous manipulation in unstructured environments, tactile sensing becomes increasingly important. In this work, we focus on object localisation through contact, where the robot must infer object state from sparse and noisy contact signals during physical interaction. This problem is particularly challenging due to continuous, high-dimensional state and action spaces, the high computational cost of physics simulation (required to accurately model contact dynamics), and the inherently discontinuous nature of contact sensor observations [6]. A common approach is to plan a sequence of "move-until-touch" actions for exploration [3], [4], [7], followed by a goal-directed phase, such as grasping. Extensions of this idea in belief-space control compute exploratory actions by optimising information-theoretic costs [5], [8], or switch between exploration and exploitation based on the belief uncertainty [9]. More principled approaches frame the task as a Partially-Observable Markov decision process (POMDP) [6], [10], [11], though tractability often requires coarse state discretisations. However, most of these methods are subject to two major limitations: *(i)* belief representations and planning typically rely on discretisation, which limits resolution and scalability; and *(ii)* object dynamics are often ignored or severely simplified, assuming static objects or restricted interaction effects. The latter is particularly problematic in contact-rich settings, where robot actions can increase uncertainty, e.g., by pushing or perturbing the object. Our work addresses both limitations by planning directly in continuous state and action spaces, and by capturing the effect of interaction dynamics on belief evolution through a non-parametric particle filter.

b) Informative Action Selection: A key challenge in planning under uncertainty is to evaluate which actions are most informative—typically by estimating their expected information gain (IG), defined as the reduction in entropy between belief states before and after an observation [3], [7], [12]. In manipulation settings, the inherent multi-modality typically requires non-parametric belief representations, such as weighted particle sets, complicating the computation of entropy. Prior work has largely relied on two approximations. One approach discretises the state space and computes discrete entropy, using occupancy grids or histograms where the entropy is defined over categorical probabilities [3]. Another approach assumes a continuous belief but approximates differential entropy using only particle weights, e.g., [9], [12], [13]. While weight-based measures reflect changes in belief confidence due to observations, they neglect the spatial distribution of particles. As a result, a belief with widely scattered particles may be assigned the same entropy as one with tightly clustered particles, despite reflecting fundamentally different levels of uncertainty. In the context of touch, this limitation becomes especially relevant: contact not only yields an observation (e.g., contact/no-contact) but can also physically alter the object's pose. Thus, touch simultaneously refines the belief and shifts the underlying state, making it essential to account for both observation-driven and dynamics-induced changes in the belief's spatial structure. In contrast, more expressive estimators of differential entropy explicitly account for the spatial distribution of the belief. For instance, kernel-based methods approximate the underlying density using a smooth distribution over particle locations [14], allowing them to capture how spread out or concentrated the belief is in state space. However, these methods are often computationally demanding and require careful tuning of kernel hyperparameters, such as bandwidth [14], [15]. [16] also derive an entropy estimate specifically for particle filters via Bayes' theorem, but this requires known transition and observation models. An alternative class of estimators based on k-nearest neighbours (kNN) density estimates provides a principled and efficient way to estimate differential entropy from samples, including for non-parametric distributions [17], [18]. Extensions to weighted particle sets make these estimators well-suited for non-parametric belief representations used in robotics [19]. These kNN density estimators can be regarded as a kernel estimator with a bandwidth that adapts to local data density [20]. Thus, kNN-based differential entropy estimators directly incorporate spatial density and have been widely adopted in reinforcement learning as intrinsic reward signals [21]–[23]. To the best of our knowledge, our work is the first to propose the use of such estimators for action selection in active tactile object localisation, where belief evolution is driven by both observations and complex interaction dynamics. This setting presents unique challenges, as actions can increase uncertainty, making spatial sensitivity essential for accurate entropy estimation. For a broader overview of non-parametric entropy estimation methods, we refer the reader to [24].

III. Problem Formulation

This paper addresses the problem of localising a rigid object o through sparse proprioceptive contact measurements using probing actions that aim to maximise the estimated expected information gain under the current belief state. We define successful localisation as reaching a state from which a grasp will succeed. To achieve this, the robot must perform information-gathering actions that reduce uncertainty about the object's location. Let $\boldsymbol{x}_t = (\boldsymbol{q}^r, \boldsymbol{q}^o) \in \mathbb{R}^{(n^r_{\mathrm{dof}}+n^o_{\mathrm{dof}})}$ describe the state of the underactuated robotic system, which includes the pose of the object $\boldsymbol{q}^o_t$, and the robot configuration $\boldsymbol{q}^r_t$ at time step t. In this work, $\boldsymbol{q}^o \in SE(2)$ denotes the object's planar pose, parameterised by (x, y, θ). The robot applies control inputs $\boldsymbol{u}_t \in \mathbb{R}^{n^r_{\mathrm{dof}}}$, which influence the object state only indirectly through contact and interac-

tion. The system evolves according to stochastic dynamics $\boldsymbol{x}_{t+1} \sim p(\cdot|\boldsymbol{x}_t, \boldsymbol{u}_t)$ and generates stochastic measurements $\boldsymbol{z}_t \sim p(\cdot|\boldsymbol{x}_t)$, where $\boldsymbol{z}_t$ represents sparse, binary contact observations. Due to the partial observability of the system and it's complex dynamics, we maintain a *continuous* belief over the system state, represented as a probability distribution $b_t{=}p(\boldsymbol{x}_t|\boldsymbol{u}_{0:t-1}, \boldsymbol{z}_{1:t}, b_0)$, which depends on the full history of actions, measurements, and the initial belief b_0. Note that we explicitly include the robot configuration in the state because it directly governs the contact interactions that drive the object's stochastic dynamics, as well as the measurements. Given a sequence of control actions $\boldsymbol{u}_{0:t-1}$, we can estimate the expected information gain $\mathrm{IG}(b_0, \boldsymbol{u}_{0:t-1})$ using dynamics and measurement models, as well as a model predicting future observations. We formulate the information gathering problem as trajectory optimisation in belief space:

$$\max_{\boldsymbol{u}_{0:T-1}} \quad \mathrm{IG}(b_0, \boldsymbol{u}_{0:T-1}), \tag{1}$$

where T is the planning horizon.

The following sections outline our approach to the information-gathering problem, with a summary illustrated in Fig. 2. In Sec. IV, we introduce a sampling-based controller that selects the next best action based on a spatially-aware estimator for the expected information gain. In Sec. V, we present our particle-based state estimator for object localisation through touch that is able to handle the discontinuous and sparse nature of the contact measurement signal. The overall approach is evaluated in Sec. VI in robot experiments and ablation studies.

Notation. In the remainder of the paper, we use subscripts for the time index and superscripts for the particle indices.

IV. Information-Gathering Controller

We propose a sampling-based controller that generates candidate probing trajectories and selects the one with the highest expected information gain (IG). The idea of sampling trajectories and ranking them by IG follows prior work in active perception [3], but here we extend it to continuous belief space and contact-rich dynamics.

A. Differential Entropy of a Weighted Particle Set

The differential entropy of a random, continuous variable $\boldsymbol{x}$ following a probability distribution $p(\boldsymbol{x})$ is defined as

$$\mathrm{H}\left[\boldsymbol{x}\right] = -\int_{\mathcal{X}} p(\boldsymbol{x}) \log p(\boldsymbol{x}) d\boldsymbol{x}. \tag{2}$$

However, the definition of differential entropy does not directly extend to finite particle sets [24]. This is due to the fact that there is no unique way to define a probability density function that is parameterised by N_p weighted samples $\{\boldsymbol{x}^i, w^i\}_{i=1}^{N_p}$. Without considering dynamics, the particle-based belief may be treated as a discrete probability distribution with the weights representing the probability of each particle. The corresponding *weight-based entropy approximation* is then given by $\hat{\mathrm{H}}_w\left[\boldsymbol{x}\right] = -\sum_i w^i \log(w^i)$ [9]. While this approximation captures information gained through observations, it does not capture the spatial density of particles, i.e., the local distance between the states of the particles. In contrast, *kernel-based* approaches approximate the underlying belief distribution b with a kernel density estimate (KDE) computed from the weighted particle set, i.e.

$$\hat{b}(\boldsymbol{x}^i) = p\left(\boldsymbol{x} \middle| \{\boldsymbol{x}^i, w^i\}_{i=1}^{N_p}\right) = \sum_i w^i \, k(\boldsymbol{x}, \boldsymbol{x}^i), \tag{3}$$

where k represents the corresponding kernel function. Fischer et al. [14] show that for particle-based belief representations the integral in the differential entropy, as defined in Eq. (2) is commonly approximated as

$$\hat{\mathrm{H}}[\hat{b}] = -\sum_{i=1}^{m} w^i \log \hat{b}(\boldsymbol{x}^i). \tag{4}$$

While this approximation accounts for the spatial density information, it is typically computationally more demanding and requires careful choice of kernel hyperparameters. In this work, we choose a uniform kernel with shared support Ω estimated from local ρ-nearest-neighbour bounding boxes for all particles (more details on how we compute Ω are provided in Appendix A. As a result, the integral in Eq. (4) simplifies to a sum over a weight-based term and a spatially-aware density term. This yields the entropy estimate

$$\hat{\mathrm{H}}[\boldsymbol{x}] \;=\; -\sum_i w^i \log w^i \;+\; \log V(\Omega), \tag{5}$$

where the first term reflects observation-driven weight changes and the second captures *spatial density* via the hypervolume $V(\Omega)$. While conceptually similar to knn-based entropy estimators [17], this formulation computes a *shared* support for all particles instead of individual supports, which is more robust to outliers and splits the entropy into two separate terms, which renders the computation of the expected information gain more efficient (cf. Sec. IV-B). The only hyperparameter is ρ for locality, and no bandwidth tuning is required. Further details on the derivation of the estimator, its properties, and a discussion of its computational complexity are provided in Appendix A.

B. Expected Information Gain

To evaluate candidate actions for information gathering, we estimate their *expected information gain (IG)*, i.e. the expected reduction in belief entropy after executing a trajectory and incorporating new observations. This process is summarised in Alg. 1. The trajectory is then truncated at the time step with maximal expected information gain.

Formally, for an initial belief b_0 and a control sequence $\boldsymbol{u}_{0:T-1}$, we define the IG as

$$\mathrm{IG}(b_0, \boldsymbol{u}_{0:T-1}) = \mathrm{H}[b_0] - \mathbb{E}_{\hat{\boldsymbol{z}}_{1:T}}\left[\mathrm{H}[b_T]\right] = \Delta \mathrm{H}_{0,T}, \tag{6}$$

where $\hat{\boldsymbol{z}}_{1:T}$ are hypothetical measurements generated along the trajectory. A common approach to estimating this expectation is to use the maximum likelihood (ML) state from the belief to simulate observations [5], [25]. However, when particle weights are uniform, such as immediately after resampling, this biases the IG estimate incorrectly towards a single hypothesis. Instead, we marginalise over all particles

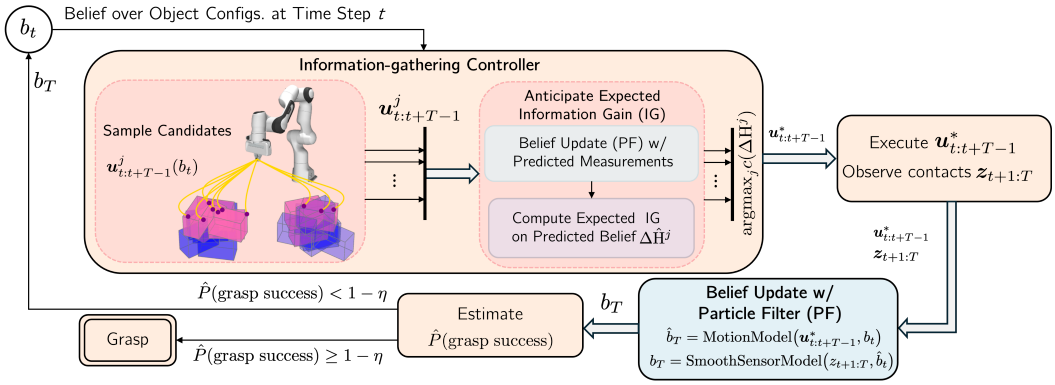


Fig. 2. Proposed pipeline for contact-rich localisation and manipulation.

in the belief to generate hypothetical measurements and compute a weighted expectation of the resulting entropy. At each time step, each particle is treated as a potential ground truth state of the system, producing a hypothetical observation from which all particle weights are updated. The expected entropy is then computed as the weighted average over all such hypotheses. Since the observations generated along the candidate trajectory are only a guess of what could occur upon execution, we estimate the information gain at time step t as if no observations have been made in previous time steps. Therefore, the update of a particle weight w^i_{t+1} is done assuming that the previous weight w^i_t is equal to w^i_0. In other words, we predict the belief states along a candidate trajectory as $b_t{=}p(\boldsymbol{x}_t|\boldsymbol{u}_{0:t-1}, \boldsymbol{z}_t, b_0)$ instead of $b_t{=}p(\boldsymbol{x}_t|\boldsymbol{u}_{0:t-1}, \boldsymbol{z}_{1:t}, b_0)$.

To compute the differential entropy for a belief state b_t, we use the estimator introduced above in Eq. (5), which decomposes the entropy into two terms: a weight-based term $\hat{\mathrm{H}}_w$ and a spatial density term $\hat{\mathrm{H}}_\Omega$. Only the weight term depends on the simulated measurements; the density term depends solely on the particle locations after propagating dynamics. Therefore we only need to marginalise the weight term over hypothetical measurements, while the density term can be computed directly from the predicted particles. The expected weight entropy at time step t is given by

$$\mathbb{E}\left[\hat{\mathrm{H}}_{w,t}\right] = \sum_j \left(-\sum_i \hat{w}^{i,j}_t \log \hat{w}^{i,j}_t \right) w^j_0, \tag{7}$$

where $\hat{w}^{i,j}_t$ is the weight of particle i at time step t when particle j is assumed to be the true state, i.e.,

$$\hat{w}^{i,j}_t = \frac{w^i_0 \; P(z_t|\boldsymbol{x}^j_t)}{\sum_l w^l_0 \; P(z_t|\boldsymbol{x}^l_t)}. \tag{8}$$

Consequently, we define the predicted entropy reduction from time step t_1 to t_2 as:

$$\Delta\hat{\mathrm{H}}_{t_1,t_2} = \left(\hat{\mathrm{H}}_{w,t_1} - \mathbb{E}[\hat{\mathrm{H}}_{w,t_2}]\right) + \left(\hat{\mathrm{H}}_{\Omega,t_1} - \hat{\mathrm{H}}_{\Omega,t_2}\right). \tag{9}$$

In our case, we always evaluate changes relative to the initial belief at $t{=}0$, i.e., $\Delta\hat{\mathrm{H}}_{0,t}$.

Algorithm 1: Predict Expected Information Gain

Input: Initial particles & weights $\{(\boldsymbol{x}^i_0, w^i_0)\}_{i=1}^{N_p}$, controls $\boldsymbol{u}_{0:T-1}$
Output: Max. expected IG $\Delta\hat{\mathrm{H}}^\star_{0,t}$ at time step $t^\star$

// Precompute constants at $t{=}0$
$\hat{\mathrm{H}}_{w,0} \leftarrow -\sum_{i=1}^{N_p} w^i_0 \log w^i_0$
$\Omega_0 \leftarrow \textsc{SharedSupport}\left(\{\boldsymbol{x}^i_0\}_{i=1}^{N_p}\right)$; $\hat{\mathrm{H}}_{\Omega,0} \leftarrow \log V(\Omega_0)$
$\Delta\hat{\mathrm{H}}^\star_{0,t} \leftarrow -\infty$; $t^\star \leftarrow 0$
for $t \leftarrow 0$ **to** $T{-}1$ **do**
  // Simulate particle dynamics under $\boldsymbol{u}_t$
  $\boldsymbol{x}^i_{t+1} \sim p(\cdot|\boldsymbol{x}^i_t, \boldsymbol{u}_t)$
  // Expected weight entropy (via Eq. (7))
  $\mathbb{E}\left[\hat{\mathrm{H}}_{w,t+1}\right] \leftarrow$
    $\textsc{ExpectedWeightEntropy}\left(\{\boldsymbol{x}^i_t, w^i_t\}_{i=1}^{N_p}, \boldsymbol{u}_t\right)$
  // Spatial entropy term
  $\Omega_{t+1} \leftarrow \textsc{SharedSupport}\left(\{\boldsymbol{x}^i_{t+1}\}_{i=1}^{N_p}\right)$
  $\hat{\mathrm{H}}_{\Omega,t+1} \leftarrow \log V(\Omega_{t+1})$
  // Predicted entropy reduction relative to $t{=}0$
  $\Delta\hat{\mathrm{H}}_{0,t+1} \leftarrow (\hat{\mathrm{H}}_{w,0} - \mathbb{E}[\hat{\mathrm{H}}_{w,t+1}]) + (\hat{\mathrm{H}}_{\Omega,0} - \hat{\mathrm{H}}_{\Omega,t+1})$
  // Keep track of max. IG and respective time step
  $(\Delta\hat{\mathrm{H}}^\star_{0,t}, t^\star) \leftarrow \max\left\{(\Delta\hat{\mathrm{H}}^\star_{0,t}, t^\star), (\Delta\hat{\mathrm{H}}_{0,t+1}, t{+}1)\right\}$
end
return $\Delta\hat{\mathrm{H}}^\star_{0,t}, t^\star$

C. Belief-Dependent Candidate Sampling

To efficiently solve the information-gathering problem in Eq. (1), it is important to generate candidate trajectories ($\boldsymbol{u}^l_{0:T-1}$) that are likely to result in contact with the object. We adopt the spline-based trajectory representation from [26] (cf. Appendix C for additional details). This representation ensures that the trajectory is smooth and kinodynamically feasible. Generating trajectories with this representation reduces to sampling the control points of the spline, which we refer to as *via-points*. There are multiple ways to design the sampling strategy for the via-points based on the current belief state, but our information-gathering controller is more effective if the sampling strategy results in a diverse set of candidate trajectories that are likely to make contact with the object. In our experiments, we show an example of how this can be achieved.

V. Particle Filter for Continuous Object Pose Estimation through Contact

The effectiveness of the information-gathering controller depends on having a reliable state estimator, particularly given the discontinuous and sparse nature of contact measurements. To estimate the belief state in the control problem of Eq.(1), we use a particle filter with N_p particles $\{\boldsymbol{x}^i, w^i\}_{i=1}^{N_p}$, where each $\boldsymbol{x}^i$ is a state sample and $w^i \in [0,1]$ is its weight, with $\sum_i w^i = 1$. The belief is updated by propagating particles through a dynamics model and re-weighting them based on measurement likelihoods, following standard particle filtering [27]. Assuming known object properties, we simulate contact dynamics using a physics engine such as MuJoCo [28]. These dynamics are deterministic given fixed initial conditions. Unlike prior work [9], [29], we do not assume the object remains static during contact.

Smooth Measurement Model from Binary Contact: The measurement model updates the belief state after applying control $\boldsymbol{u}_t$ and observing measurement $\boldsymbol{z}_t$. We infer binary contact (contact or no-contact) from proprioceptive torque signals on the robot joints (see Appendix D). However, modeling contact as a strict binary process, as in [3], is limiting as it does not account for the fact that the robot might be close to the object without actually making contact. In addition, unlike prior work [10], [29], we do not assume that contact measurements are perfectly discriminative. To smooth this discontinuous signal, we introduce a proximity-based measurement model where the probability of contact decays exponentially with the robot-object distance. Specifically, the likelihood of contact given particle pose $\boldsymbol{x}_k^i$ is:

$$P(z_t = 1|\boldsymbol{x}_t^i) = \alpha_{\text{fp}} + (\alpha_{\text{tp}} - \alpha_{\text{fp}}) \exp(-\gamma\, d(\boldsymbol{q}_t^r, \boldsymbol{q}_{i,t}^o)), \quad (10)$$

where α_{tp} and α_{fp} are the true/false positive contact rates, γ is a decay rate, and $d(\cdot)$ is the minimum Euclidean distance between robot and object configurations (not limited to the end-effector, unlike [30]). The no-contact likelihood is $1 - P(z_t = 1|\boldsymbol{x}_t^i)$. This smoothing improves belief tracking and reduces particle starvation, but adds computational overhead due to distance evaluations. As this reduces but not eliminates the likelihood of particle starvation, we also introduce a proximity-aware resampling strategy that replaces low-weight particles with samples better matching the latest observation, while using standard resampling for no-contact cases (cf. Appendix D).

VI. Experiments

In order to evaluate all core components of our proposed framework, we conduct two sets of experiments. First, we demonstrate the proposed framework in a robot setup for object localisation and subsequent grasping through sparse contact measurements in Sec. VI-A. Second, we conduct an ablation study to evaluate the impact of the proposed smooth measurement model and resampling strategy on the belief tracking performance in Sec. VI-B.

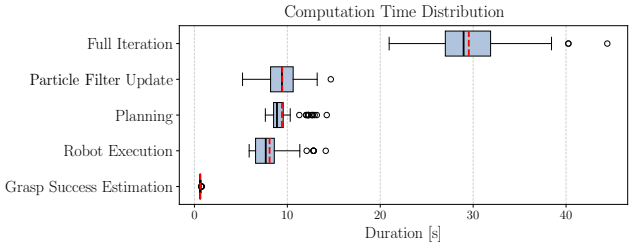


Fig. 3. Overview of computation times for one full iteration and the different sub-steps of the algorithmic framework, as shown in Fig.2, across experiments. Boxplots for the sub-steps are sorted from the longest average duration to the shortest from top to bottom. Dashed red lines correspond to the mean, and solid black lines to the median.

A. Robot Experiments

a) Setup: We use a Franka Emika robot manipulator to localise and grasp a box in its workspace (see Fig. 1). The binary contact measurements are computed from the measured external torques acting on the robot joints (see Appendix D).We show an overview of the different steps in the proposed system consisting of the information-gathering controller and the state estimation in Fig. 2. Given only an initial set of particles sampled from a bi-modal Gaussian mixture distribution, we repeatedly sample candidate trajectories, estimate their information gain, execute the highest information gain trajectory, update the particle filter belief and finally estimate the probability of successfully grasping the box given the updated belief (cf. Appendix D). If the estimated probability of success is above a given threshold, the robot attempts an open-loop grasp of the object. In order to ensure that we generate candidate trajectories that are likely to result in contact events, we sample the via-points such that the corresponding trajectories poke into the object belief. More precisely, we generate the final via-points by importance sampling particles from the current belief. Then, for each particle, we uniformly sample a point within the object's volume and compute the inverse kinematics solution such that the robot's end-effector reaches the sampled point. We constrain our motions to poking due to the unreliability of the external torque measurements, which makes it difficult to distinguish between contact and no-contact when the robot pushes an object over longer distances. For each experiment, we report the grasp success rate (with 95% Wilson confidence intervals) and the mean number of iterations required to achieve a successful grasp. The number of iterations is limited to 15. We run each experiment 30 times with random initial box poses. More implementation details for the robot experiments can be found in the Appendix D.

b) Baselines: We compare our information-gathering controller against two baselines: *i) Uninformed baseline:* the framework described above but without an information-gain metric to choose trajectories; and *ii) Maximum contact baseline:* the framework described above, but that chooses trajectories that maximise the number of particles contacted in the belief state.

c) Results: The results, summarised in Table I, show that the proposed method significantly outperforms both baselines in terms of success rate and efficiency. A compila-

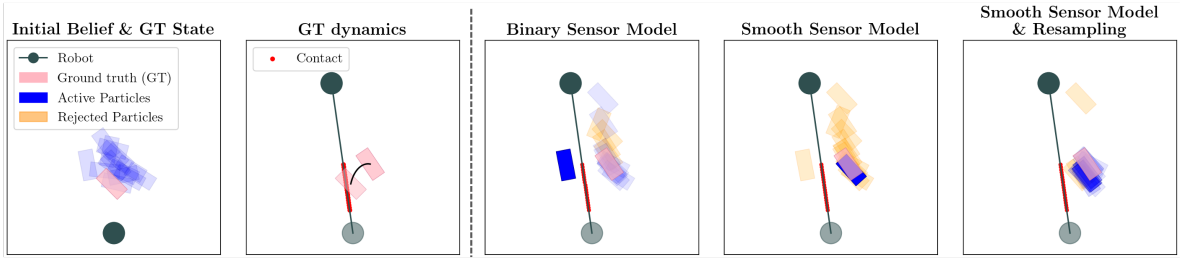


Fig. 4. *Ablation Study of Sensor Models and Resampling Strategy.* We use the following setup to compare three particle-filter variants. A circular robot (dark green) pushes a rectangular object (pink) along a fixed straight-line trajectory (light green) in a planar *quasi-static* setting. The left-most panel shows the initial particle distribution and ground-truth (GT) state; the next panel shows the GT rollout with contact events in red. To the right of the dashed line, we compare the binary sensor model [3], our smooth sensor model, and our smooth model in combination with contact-aware resampling. The smooth model with resampling best preserves particle diversity and while still accurately tracking the belief.

TABLE I

ROBOT EXPERIMENTS: PROPOSED APPROACH COMPARED TO BASELINES

Models	Success rate ↑	Avg. num. iterations (± std.) ↓
Proposed	**0.8** [0.63, 0.90]	9.37 ±3.47
Uninformed	0.23 [0.12, 0.41]	12.53 ±3.68
Max. Contact	0.33 [0.19, 0.51]	13.77 ±2.76

TABLE II

ABLATION RESULTS OF PARTICLE FILTER

Method	MSE ↓		Active Particles ↑ (%)
	Position	Orientation	
Binary	0.023 ± 0.01	0.208 ± 0.09	0.038 ± 0.02
Smooth	0.019 ± 0.01	0.185 ± 0.10	0.077 ± 0.03
Smooth & Resampled	**0.018** ± 0.01	**0.164** ± 0.17	**0.511** ± 0.14

tion of all experiments, showcasing our approach alongside the baselines, is provided in the accompanying video[1]. Notably, for the given object, the margin of error in the object localisation, allowing for a successful grasp, is very small, as the end-effector width is slightly larger than the object width. We also report the computation times spent on different parts in the algorithmic pipeline across iterations of the algorithm and experiment runs for our approach in Fig. 3.

B. Ablation Studies

We compare three particle-filter variants: (i) the binary sensor model of [3], (ii) our smooth sensor model, and (iii) the smooth model with resampling. We evaluate their performance in a planar, quasi-static simulation with a circular robot and a rectangular object with each run using 100 particles and a straight-line robot trajectory (see the qualitative example in Fig. 4). We report the *Mean Squared Error (MSE)* ± std over 1,000 runs in position and orientation computed over the top-10 weighted particles, and the final *share of active particles (AP)*, defined as the fraction of particles with weights $> 10^{-4}$ (Table II). Relative to the binary baseline, the smooth model yields better position estimates, and adding resampling further improves position accuracy. While resampling slightly worsens orientation error, it substantially increases particle diversity, consistent with the qualitative example in Fig. 4. The share of active particles is highest with resampling. Note that we do not resample at the final step to obtain a realistic estimate of particle diversity; immediately after resampling, the share of active particles would trivially be one.

[1]Note that the AR-tag in the videos was only used to add the initial and final ground truth object position to the Mujoco renderings in the videos.

VII. LIMITATIONS

We acknowledge several limitations and outline promising future directions. First, the applicability of our method to replanning is limited by the computational cost of rolling out candidate control trajectories in a physics engine. Replacing this with a learned contact dynamics model from real-world data could reduce the current computational complexity, while also eliminating the need for known object properties (e.g., mass, friction), and better capturing the inherent stochasticity of contact interactions. Second, our current setup assumes an uncluttered scene and uses sparse binary contact signals. Richer tactile sensing would enable detection of lighter or more diverse objects. Future work will consider more complex, cluttered environments and varied object geometries. While our approach is not tied to specific geometries, sim-to-real discrepancies may grow with non-rigid or more complex objects. Incorporating residual dynamics learning could enhance robustness to these challenges. Lastly, a compelling extension would be to broaden the information-theoretic framework beyond sensing-driven actions to include actions that actively funnel uncertainty into smaller regions of the state space, as explored in prior work [31]–[33].

VIII. CONCLUSION

We presented a continuous-state, touch-only localisation framework capable of operating under high initial uncertainty. Our system combines a proximity-aware contact model with contact-aware resampling to maintain informative, non-parametric particle beliefs during physical interaction. A central component of our approach is the use of a k-nearest-neighbour-based entropy estimator that accounts for both observation-driven weight changes and interaction-driven shifts in particle distribution. This enables effective evaluation of candidate actions based on how they reshape the belief through both sensing and dynamics. To the best of our knowledge, this is the first use of knn-based estimators for action selection in active tactile object localisation, where

the belief evolution is fundamentally shaped by contact and physical interaction. We validate our method in simulation and on real hardware, demonstrating reliable localisation and grasping under large, multi-modal uncertainty regions, far beyond those addressed in prior work. Compared to baselines relying on heuristic contact maximisation or weight-only entropy, our approach yields higher success rates and more efficient exploration. These results highlight the importance of explicitly modelling both the sensory and physical structure of contact for information-driven manipulation.

REFERENCES

[1] N. Fazeli, M. Oller, J. Wu, Z. Wu, J. B. Tenenbaum, and A. Rodriguez, "See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion," *Science Robotics*, vol. 4, no. 26, 2019.

[2] A. Rodriguez, "The unstable queen: Uncertainty, mechanics, and tactile feedback," *Science Robotics*, vol. 6, no. 54, 2021.

[3] P. Hebert, T. Howard, N. Hudson, J. Ma, and J. W. Burdick, "The next best touch for model-based localization," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 99–106.

[4] A. Petrovskaya and O. Khatib, "Global localization of objects via touch," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 569–585, 2011.

[5] R. Platt, L. Kaelbling, T. Lozano-Perez, and R. Tedrake, "Efficient planning in non-gaussian belief spaces and its application to robot grasping," in *Robotics Research*. Springer, 2017, pp. 253–269.

[6] M. C. Koval, N. S. Pollard, and S. S. Srinivasa, "Pre-and post-contact policy decomposition for planar contact manipulation under uncertainty," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 244–264, 2016.

[7] S. Javdani, M. Klingensmith, J. A. Bagnell, N. S. Pollard, and S. S. Srinivasa, "Efficient touch based localization through submodularity," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 1828–1835.

[8] R. Platt, L. Kaelbling, T. Lozano-Perez, and R. Tedrake, "Simultaneous localization and grasping as a belief space control problem," in *International Symposium on Robotics Research*, vol. 2, 2011.

[9] E. Nikandrova, J. Laaksonen, and V. Kyrki, "Towards informative sensor-based grasp planning," *Robotics and Autonomous Systems*, vol. 62, no. 3, pp. 340–354, 2014.

[10] K. Hsiao, L. P. Kaelbling, and T. Lozano-Pérez, "Task-driven tactile exploration." in *Robotics: science and systems*, vol. 12, 2010.

[11] M. Horowitz and J. Burdick, "Interactive non-prehensile manipulation for grasping via pomdps," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3257–3264.

[12] B. Saund, S. Chen, and R. Simmons, "Touch based localization of parts for high precision manufacturing," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 378–385.

[13] N. Miyazawa, D. Kato, Y. Kobayashi, K. Hara, and D. Usui, "Optimal action selection to estimate the aperture of bag by force-torque sensor," *Advanced Robotics*, vol. 38, no. 2, pp. 95–111, 2024.

[14] J. Fischer and Ö. S. Tas, "Information particle filter tree: An online algorithm for pomdps with belief-based rewards on continuous domains," in *International Conference on Machine Learning*. PMLR, 2020, pp. 3177–3187.

[15] B. W. Silverman, *Density estimation for statistics and data analysis*. Routledge, 2018.

[16] Y. Boers, H. Driessen, A. Bagchi, and P. Mandal, "Particle filter based entropy," in *2010 13th International Conference on Information Fusion*. IEEE, 2010, pp. 1–8.

[17] H. Singh, N. Misra, V. Hnizdo, A. Fedorowicz, and E. Demchuk, "Nearest neighbor estimates of entropy," *American journal of mathematical and management sciences*, vol. 23, no. 3-4, pp. 301–321, 2003.

[18] J. Jiao, W. Gao, and Y. Han, "The nearest neighbor information estimator is adaptively near minimax rate-optimal," *Advances in neural information processing systems*, vol. 31, 2018.

[19] J. Ajgl and M. Šimandl, "Differential entropy estimation by particles," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 11 991–11 996, 2011.

[20] T. B. Berrett, R. J. Samworth, and M. Yuan, "Efficient multivariate entropy estimation via k-nearest neighbour distances," 2019.

[21] H. Liu and P. Abbeel, "Behavior from the void: Unsupervised active pre-training," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 459–18 473, 2021.

[22] Y. Seo, L. Chen, J. Shin, H. Lee, P. Abbeel, and K. Lee, "State entropy maximization with random encoders for efficient exploration," in *International conference on machine learning*. PMLR, 2021, pp. 9443–9454.

[23] M. Mutti, L. Pratissoli, and M. Restelli, "Task-agnostic exploration via policy gradient of a non-parametric state entropy estimate," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, 2021, pp. 9028–9036.

[24] J. Beirlant, E. J. Dudewicz, L. Györfi, E. C. Van der Meulen *et al.*, "Nonparametric entropy estimation: An overview," *International Journal of Mathematical and Statistical Sciences*, vol. 6, no. 1, pp. 17–39, 1997.

[25] R. Platt, R. Tedrake, L. P. Kaelbling, and T. Lozano-Perez, "Belief space planning assuming maximum likelihood observations," in *Robotics: Science and Systems*, 2010.

[26] J. Jankowski, L. Brudermüller, N. Hawes, and S. Calinon, "Vp-sto: Via-point-based stochastic trajectory optimization for reactive robot behavior," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 125–10 131.

[27] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, ser. Intelligent Robotics and Autonomous Agents series. MIT Press, 2005.

[28] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.

[29] M. C. Koval, N. S. Pollard, and S. S. Srinivasa, "Pose estimation for planar contact manipulation with manifold particle filters," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 922–945, 2015.

[30] C. Zito, V. Ortenzi, M. Adjigble, M. Kopicki, R. Stolkin, and J. L. Wyatt, "Hypothesis-based belief planning for dexterous grasping," *arXiv preprint arXiv:1903.05517*, 2019.

[31] M. A. Erdmann and M. T. Mason, "An exploration of sensorless manipulation," *IEEE Journal on Robotics and Automation*, vol. 4, no. 4, pp. 369–379, 1988.

[32] M. R. Dogar, K. Hsiao, M. T. Ciocarlie, and S. S. Srinivasa, "Physics-based grasp planning through clutter." in *Robotics: Science and systems*, vol. 8, 2012, pp. 57–64.

[33] J. Jankowski, L. Brudermüller, N. Hawes, and S. Calinon, "Planning for robust open-loop pushing: Exploiting quasi-static belief dynamics and contact-informed optimization," *arXiv preprint arXiv:2404.02795*, 2024.

6.1 Differential Entropy of Non-Parametric Beliefs

This section expands upon the particle-based differential entropy estimator for weighted particle sets introduced in Section IV-A of the main paper. Previously presented in the Appendix, this content is now integrated into the main body of the chapter to facilitate a conceptual comparison with existing kNN-based entropy estimators from the literature.

Estimator We consider a belief represented by a set of particles $\{\boldsymbol{x}^i\}_{i=1}^N$ with associated weights $\{w^i\}_{i=1}^N$. To estimate the differential entropy of a particle-based belief, we approximate the belief as a mixture of uniform distributions, each centred at a particle $\boldsymbol{x}^i$ with weight w^i and shared support $\Omega \subseteq \mathcal{X}$, a bounded region in state space with hypervolume $V(\Omega) = \int_\Omega 1 d\boldsymbol{x}$:

$$\hat{b}(\boldsymbol{x}) = \sum_i w^i \, \mathcal{U}(\boldsymbol{x} - \boldsymbol{x}^i, \Omega), \tag{6.1}$$

where $\mathcal{U}(\boldsymbol{x}, \Omega)$ denotes the uniform density over Ω, i.e. $\mathcal{U}(\boldsymbol{x}, \Omega) = {}^1\!/_{V(\Omega)}$ if $\boldsymbol{x} \in \Omega$ and zero otherwise. The differential entropy of such a belief can be approximated as

$$\hat{\mathrm{H}}\,[\boldsymbol{x}] = -\sum_i w^i \log w^i + \log V(\Omega), \tag{6.2}$$

where the hypervolume $V(\Omega)$ encodes the local density of the particles. In the following, we will use $V(\Omega)$ and V interchangeably. We determine the *shared* support Ω as follows: *i)* For each particle i, fit a tight bounding box ${}^\rho\Omega_i$ to its ρ-nearest neighbors, *ii)* compute the average bounding box ${}^\rho\bar{\Omega}$ by averaging box dimensions, and *iii)* scale it to obtain $\Omega = \frac{{}^\rho\bar{\Omega}}{\left(\sqrt[d]{\rho}-1\right)}$, where d is the number of dimensions. This procedure captures local particle density, i.e., higher densities lead to smaller $V(\Omega)$ and lower entropy. The hyperparameter ρ controls the locality of the approximation: small values capture multimodality; large values yield global smoothing. We also illustrate the computation of Ω in Fig. 6.1. Thus, Eq. (6.2)

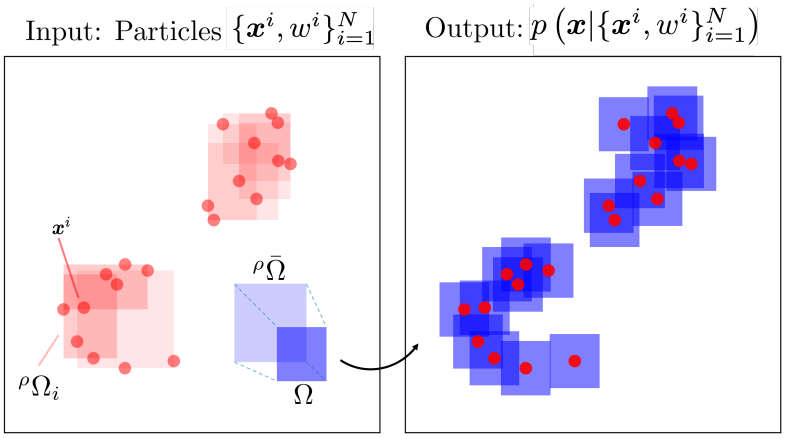


Figure 6.1: *Computation of the shared support* Ω that is used to construct the probability density function p for a set of weighted particles (●). After fitting a tight bounding box ${}^{\rho}\Omega_i$ (■) to the $\rho = 6$ nearest neighbours of each particle, we scale the mean bounding box ${}^{\rho}\bar{\Omega}$ to obtain the shared support Ω (■) that is used across all particles.

can be interpreted as consisting of two parts:

$$\hat{\mathrm{H}}\left[\boldsymbol{x}\right] = \underbrace{-\sum_i w^i \log w^i}_{\hat{\mathrm{H}}_w[\boldsymbol{x}]} + \underbrace{\log V(\Omega)}_{\hat{\mathrm{H}}_\Omega[\boldsymbol{x}]},$$

where $\hat{\mathrm{H}}_w[\boldsymbol{x}]$ depends only on the weight distribution and $\hat{\mathrm{H}}_\Omega[\boldsymbol{x}]$ captures the local density encoded by the shared support.

Computational Complexity The main cost is computing ρ-nearest neighbors for N_p particles, with complexity $\mathcal{O}(N_p D \log(N_p + \rho))$ using a ball tree. This is more efficient than kernel density estimation with complexity $\mathcal{O}(N_p^2 D^3)$ (Fischer and Tas, 2020).

Limitation The differential entropy may diverge to $-\infty$ if the distribution collapses in any dimension, which corresponds to $V(\Omega) \to 0$. To mitigate this, we compute entropy only over relevant dimensions (e.g., omitting z-dimension if particles lie on a table).

6.1.1 Upper Bound on the Entropy of a Uniform Mixture Distribution

We now show that the estimator in Eq. (6.2) provides an *upper bound* on the true differential entropy of the uniform mixture distribution defined in Eq. (6.1).

Derivation of the Upper Bound The differential entropy of a continuous random variable $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ with density $p(\boldsymbol{x})$ is

$$\mathrm{H}[\boldsymbol{x}] = -\int_{\mathcal{X}} p(\boldsymbol{x}) \log p(\boldsymbol{x})\, d\boldsymbol{x}. \tag{6.3}$$

For the mixture distribution in Eq. (6.1), this becomes

$$\mathrm{H}[\boldsymbol{x}] = -\int_{\mathcal{X}} \Big(\sum_i w^i \mathcal{U}_i(\boldsymbol{x}) \Big) \log\Big(\sum_j w^j \mathcal{U}_j(\boldsymbol{x}) \Big)\, d\boldsymbol{x}, \tag{6.4}$$

where we denote $\mathcal{U}_i(\boldsymbol{x}) = \mathcal{U}(\boldsymbol{x} - \boldsymbol{x}^i, \Omega)$. Pulling the first sum out of the integral yields

$$\mathrm{H}[\boldsymbol{x}] = -\sum_i w^i \int_{\mathcal{X}} \mathcal{U}_i(\boldsymbol{x}) \log\Big(\sum_j w^j \mathcal{U}_j(\boldsymbol{x}) \Big)\, d\boldsymbol{x}. \tag{6.5}$$

Since $\mathcal{U}_i(\boldsymbol{x})$ has support $\Omega_i = \Omega - \boldsymbol{x}^i$, this simplifies to

$$\mathrm{H}[\boldsymbol{x}] = -\sum_i w^i \int_{\Omega_i} \frac{1}{V} \log\Big(\sum_j w^j \mathcal{U}_j(\boldsymbol{x}) \Big) d\boldsymbol{x}. \tag{6.6}$$

The integral in Eq. (6.6) cannot be evaluated in closed form due to possible overlaps of the uniform components. To bound it, we use the inequality

$$\log\Big(\sum_j w^j \mathcal{U}_j(\boldsymbol{x}) \Big) \;\geq\; \log\Big(w^i \tfrac{1}{V} \Big), \tag{6.7}$$

which holds for all $\boldsymbol{x} \in \Omega_i$. Equality arises if the supports of different components do not overlap, i.e., $\Omega_i \cap \Omega_j = \emptyset, \forall i \neq j$. Substituting Eq. (6.7) into Eq. (6.6) gives

$$\mathrm{H}[\boldsymbol{x}] \leq -\sum_i w^i \int_{\Omega_i} \tfrac{1}{V} \Big(\log w^i + \log \tfrac{1}{V} \Big) d\boldsymbol{x}. \tag{6.8}$$

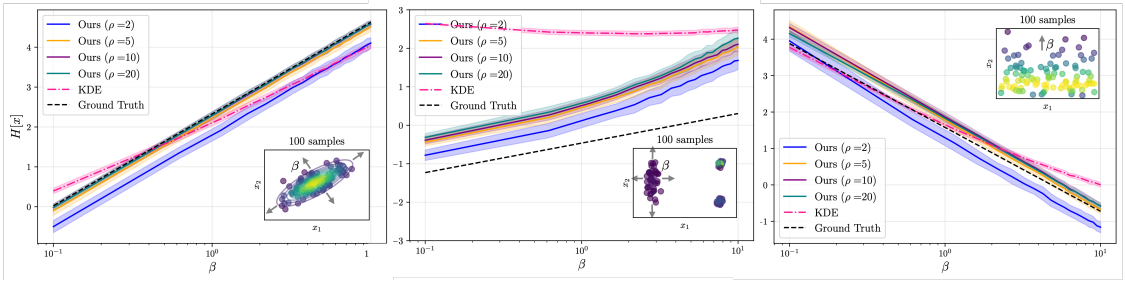
Since the integrand is constant over Ω_i, the integral resolves directly, yielding

$$\mathrm{H}[\boldsymbol{x}] \leq -\sum_i w^i \Big(\log w^i + \log \tfrac{1}{V} \Big). \tag{6.9}$$

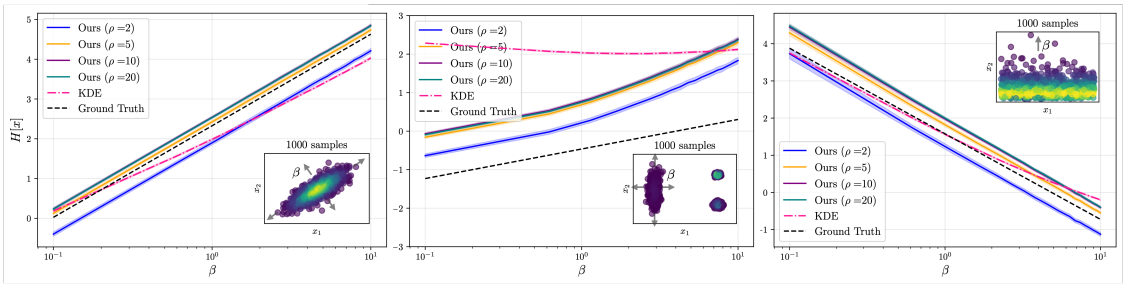
Finally, using $\sum_i w^i = 1$, we arrive at

$$\mathrm{H}[\boldsymbol{x}] \leq -\sum_i w^i \log w^i + \log V, \tag{6.10}$$

which matches the estimator in Eq. (6.2), thereby confirming it as an upper bound on the true entropy.



(a) Entropy approximations using 100 particles.



(b) Entropy approximations using 1000 particles.

Figure 6.2: ***Comparison of the proposed particle-based entropy estimator*** (Eq. (6.2)) for different numbers of nearest neighbours ρ used to compute the shared support Ω, against *(i)* a KDE-based approximation with a multivariate Gaussian kernel and *(ii)* the analytical differential entropy, evaluated on three different ground-truth probability distributions. All approximations are computed on a set of either (a) 100 or (b) 1000 particles sampled from the respective distribution. Parameter β scales the entropy of the ground truth distributions, shown in the inset plots for β=1, as follows: *Left:* Gaussian distribution with covariance matrix scaled by β. *Center:* Gaussian mixture distribution with two fixed components on the right-hand side and one scaled component on the left-hand. All components do not overlap. *Right:* Distribution of two independent variables, x_1 following a uniform distribution and x_2 a Gamma distribution with the rate parameter set to β.

Quantitative Analysis of the Approximation Quality of the Entropy Estimate We perform a quantitative analysis of the estimator, shown in Fig. 6.2, for three theoretical ground truth distributions that have an analytic solution for the differential entropy in Eq. (6.3). We compare the analytic solution against the estimator in Eq. (6.2), denoted as "ours", with varying numbers of ρ neighbours in the computation of the shared support Ω against a kernel-based (KDE) entropy approximation, defined as

$$\hat{\mathrm{H}}[\boldsymbol{x}] = -\sum_{i=1}^{m} w^i \log\Big(\sum_i w^i \, k(\boldsymbol{x}, \boldsymbol{x}^i)\Big). \tag{6.11}$$

We use a multivariate Gaussian kernel k with the bandwidth matrix tuned according to Silverman's rule of thumb (Silverman, 2018; Fischer and Tas, 2020). We evaluate the approximations on a set of either 100 or 1000 particles sampled from the respective distribution across 100 runs. Our estimator (Eq. (6.2)) outperforms the kernel-based approximation in Eq. (6.11) in terms of overall accuracy. Regarding the hyperparameter ρ, the GMM example shows that using a lower number of nearest neighbours in the support computation is able to better capture the local densities of the distribution. Last, we note that while the approximation for a newly sampled set of particles can be noisy, it still captures the monotonic trend of the ground truth entropy. Even more importantly, for reducing uncertainty, the approximation is deterministic and thus captures small changes to a given set of particles.

6.2 Limitations and Future Work

Computational Efficiency One limitation of the current approach, as also mentioned in the main paper, is its computational efficiency. We have shown an overview of the computation times in Figure 3 in the paper. To elaborate further, we see two main bottlenecks: *(i)* the rollout of the belief dynamics, in particular when computing the entropy estimates of the sampled trajectories, and *(ii)* the distance computations in the smooth sensor model and the resampling strategy. For *(i)*, we believe that an interesting direction for future work would be to explore learned belief dynamics models, such as recently presented by (Tremblay et al., 2025), which could potentially speed up the rollouts significantly. For *(ii)*, besides potential hardware improvements, i.e. using richer sensors that would eliminate the need for smoothing, we could also explore learned particle resampling strategies, e.g. using learned models to predict particles along the difficult to model contact manifold (Röstel et al., 2022). In addition, we could explore learning a generative sensor model (Chen and Li, 2023), instead of the full belief dynamics, which would equally remove the need for distance computations. A promising approach could, for instance, be to combine the learned state estimation approach from (Röstel et al., 2022) with our information-gathering control framework.

Solving the Underlying POMDP While our work presents a sampling-based approach to belief space control, it does not solve the underlying POMDP in a principled manner. Future work could explore solving the underlying POMDP more explicitly. While traditional POMDP solvers are not directly applicable due to the continuous and high-dimensional nature of the belief space, recent advances in Monte Carlo Tree Search (MCTS) for continuous spaces (Sunberg and Kochenderfer, 2018) could be adapted to our setting. The discontinuities in belief dynamics caused by contact-based observations could then be managed using the approaches proposed here, or by learning a smooth surrogate of the belief dynamics, as outlined above. The algorithm could then be structured in a similar fashion to *MuZero* (Schrittwieser et al., 2020) or *BetaZero* (Moss et al., 2023), where online Monte Carlo Tree Search is sped up by learning from experience in an offline policy iteration step that learns neural network surrogates of the value function and the action-selection policy. Yet, both approaches still assume that propagating the belief state through the search tree is computationally inexpensive, which is not the case in our setting, where the belief state is continuous and high-dimensional. Hence, again learned surrogates of the belief state dynamics could be used to propagate the belief state through the search tree.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Touch-based object localisation with spatially-aware belief entropy estimation	
Publication Status	□ Published	□ Accepted for Publication
	☑ Submitted for Publication	□ Unpublished and unsubmitted work written in a manuscript style
Publication Details	Brudermüller, L., Jankowski, J., Toussaint, M., and Hawes, N. (2025c). Touch-based object localisation with spatially-aware belief entropy estimation. Manuscript under review at IEEE International Conference on Robotics and Automation (ICRA)	

Student Confirmation

Student Name:	Lara Brudermüller		
Contribution to the Paper	– I led the development of the idea in collaboration Julius Jankowski, Marc Toussaint and Nick Hawes. – I implemented the algorithm and all of the experiments. – I conducted all experiments with support from Marc Toussaint and Julius Jankowski. – I led the writing and incorporated feedback from Julius Jankowski, Marc Toussaint and Nick Hawes.		
Signature		Date	September 17, 2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Professor Nick Hawes	
Supervisor comments I confirm that Lara made a substantial contribution to this publication, and that the description above is accurate.		
Signature		Date: September 17, 2025

This completed form should be included in the thesis, at the end of the relevant chapter.

7
Conclusions

In this thesis, we presented theoretical and algorithmic contributions towards robot manipulation under uncertainty. Contact-rich manipulation presents unique challenges for planning and control due to the inherent uncertainty in contact dynamics, the high dimensionality of the state and action spaces, and the discontinuities introduced by contact events. Addressing these challenges requires methods that can effectively handle uncertainty, adapt to changing environments, and efficiently explore complex solution spaces. We proposed four complementary methods that address different facets of planning and control for robot manipulation under uncertainty, unified by a reliance on sampling-based optimisation as a robust, general-purpose tool for handling discontinuities and non-smoothness in dynamics and cost functions. A central insight from this thesis is that the uncertainty inherent in contact-rich manipulation can be addressed through a combination of *implicit* and *explicit* strategies, depending on the nature and context of the uncertainty.

Implicit Uncertainty Handling In scenarios where uncertainty is predominantly *aleatoric* and arises from sensor noise or unpredictable contact events, frequent replanning, enabled by sampling-based MPC, can effectively mitigate its impact. We referred to methods presented in this context as *implicit* approaches to uncertainty handling, as they do not explicitly model or reason about uncertainty,

but rather rely on the controller's ability to adapt quickly to new information. In Chapter 3, we built on the insight that stochastic optimisation methods, such as CMA-ES, provide a powerful means of addressing the discontinuous and non-convex optimisation landscapes induced by contact dynamics. We proposed a new trajectory parameterisation that maps a small set of via-points to continuous, timing-optimal motions, enabling the incorporation of informative probabilistic priors, such as smoothness and a contact-making bias, directly into the optimisation process. These priors facilitated efficient exploration of the contact space, allowing the robot to make and break contacts in real time, even in the presence of unmodelled interactions and abrupt environmental changes. While this approach has been shown to be effective in real-time contact-rich control, it remains limited in two important ways. First, the underlying sampling process relies on Gaussian distributions, which constrains the expressiveness of the trajectory proposals and, consequently, the diversity of solutions that could be explored during optimisation. Second, because of the shooting-based formulation, the computational efficiency of the approach is fundamentally limited by the need for forward simulations of complex contact dynamics, restricting both the planning horizon and the number of samples that can be evaluated within the time budget available for online control. To overcome these limitations, Chapter 4 explored how more expressive sampling distributions can be learned from offline planner data. In particular, we trained generative models on control sequences collected from running sampling-based MPC methods with longer horizons, larger sample sizes and without real-time constraints. This enables fast online control that retains the benefits of sampling-based optimisation while leveraging richer, data-driven proposal distributions. We demonstrated that this approach can solve complex loco-manipulation tasks that require both expensive forward simulations and long planning horizons, which are infeasible with standard sampling-based MPC methods. Together, these two chapters highlighted the effectiveness of implicit uncertainty handling through reactive control with sampling-based MPC methods.

Explicit Uncertainty Handling In scenarios where uncertainty is predominantly *epistemic*, arising from partial observability, unmodelled dynamics, or unknown object properties, reactive control alone is insufficient. In such cases, a robot must *explicitly* reason about uncertainty, anticipating its effects on future outcomes and, when necessary, acting to actively reduce it. This requires methods that can model uncertainty, predict its evolution over time, and incorporate it into the planning process. To this end, we extended the VP-STO framework from Chapter 3 to account for stochastic system dynamics through chance constraints in Chapter 5. A tractable sample-efficient approximation of chance constraints allowed the planner to generate motions that satisfy constraints with high probability, while still being real-time capable. This approach lets the planner balance task performance with robustness to uncertain outcomes, leading to more reliable and efficient behaviour in environments with stochastic dynamics. While we demonstrated the approach in the context of dynamic obstacle avoidance, the method is broadly applicable to a wide range of robotic systems and tasks, including contact-rich manipulation, making no assumptions about the specific structure of uncertainty beyond sampling-based access to the underlying distributions. In contrast, Chapter 6 focused on *state uncertainty*, proposing a belief-space control approach for global object localisation through contact-based exploration. The key insight here is that information-theoretic objectives, such as entropy, if accounting for contact interactions, can provide offer a principled means to select actions that reduce uncertainty in partially observable settings in order to achieve a downstream task.

Bridging Reactive and Robust Control Together, these contributions highlight a central insight of this thesis: robust and general-purpose robot manipulation arises from integrating both *implicit* and *explicit* approaches to uncertainty handling. Relying solely on reactive replanning limits foresight, while reasoning about uncertainty without reactivity constrains adaptability. By bridging these paradigms, this thesis demonstrates that sampling-based optimisation naturally accommodates both reactive control and explicit consideration of uncertainty, linking real-time

adaptability with robustness to uncertainty. This integration moves us closer to a cohesive framework for robot manipulation that is capable of operating reliably in the complex, uncertain environments that define real-world manipulation.

7.1 Future Work

As with all research, this thesis opens up many avenues for future exploration. We have already discussed several limitations and future directions in the respective chapters. Below, building on the foundations laid in this thesis, we outline some broader directions that draw connections between multiple chapters and that we consider particularly promising towards the ultimate goal of robust and general-purpose robot manipulation in unstructured, uncertain environments. While we have demonstrated methods that can handle uncertainty in contact-rich manipulation, the respective demonstrations have so far still been limited to controlled lab settings. Yet, the ultimate goal is to make these methods work in truly unstructured environments, such as homes, offices, and public spaces. Imagine a robot helping us cook dinner: it needs to fetch ingredients from cluttered shelves, open jars and bottles, use utensils to prepare food, and clean up afterwards. This requires a robot to manipulate a wide variety of objects, often in contact-rich ways, while dealing with significant uncertainty about the environment, object properties, and its own state. Some of the objects may be partially occluded, deformable, or articulated, making it challenging to perceive and model them accurately. A robot will also need to adapt to dynamic changes in the environment, such as moving objects or people, and handle unexpected events, such as spills or dropped items. We believe that achieving this level of autonomy will require combining the strengths of the methods presented in this thesis with advances in other areas, such as world models, and state representation learning. Below, we discuss these directions in more detail.

World Models The core limitations of the model-based methods presented in this thesis are the reliance on full-state information and the computational burden of forward simulations with physics-based models. Learning-based dynamics

models (Ai et al., 2025) show great potential to addressing these limitations. Traditional physics-based models, while interpretable and grounded in first principles, often fail to capture the full complexity of real-world dynamics, especially in contact-rich scenarios where frictional interactions, compliance, and unmodelled effects play a significant role, also known as simulation-to-reality (sim-to-real) gap (Zhao et al., 2020). Moreover, most of these models are limited to rigid-body dynamics, while extending them to deformable or articulated systems introduces substantial additional complexity and computational cost. Trained directly from interaction or sensory data, learned dynamics models offer a promising alternative to analytical models by implicitly capturing real-world dynamics. Such models can reduce reliance on precise system identification, compensate for state estimation errors, and in some cases bypass state estimation altogether by learning latent dynamics from raw sensory inputs (Hafner et al., 2019). Incorporating these data-driven approaches may enhance robustness, improve sim-to-real transfer, and enable more adaptive and computationally efficient control frameworks. While world models have shown great promise in model-based reinforcement learning (Li et al., 2025; Wu et al., 2023; Hafner et al., 2025), their integration with MPC has been less explored, with some notable exceptions (Hansen et al., 2022; Zhou et al., 2024; Huang et al., 2025; Jain et al., 2025). For the methods presented in this thesis, learned dynamics models could not only help reducing the computational burden of forward simulations, but could also enable planning directly from raw sensory inputs. However, besides their potential, they also introduce new challenges, such as model bias and compounding errors over long horizons. Addressing these issues requires careful model design, uncertainty quantification methods (e.g., conformal prediction (Sun et al., 2023)), and integration with robust planning methods, such as those presented in this thesis. In addition, besides solely using fixed learned models for planning, an interesting direction would be to explore online adaptation of learned models, e.g. through online system identification (Xu et al., 2019; Ai et al., 2024). The methods presented in Chapter 6 could be particularly useful in

this context, as they provide a means to actively reduce uncertainty about model parameters through contact-based exploration.

State Representation Learning Most methods presented in this thesis assume access to full-state information, which is often not available in real-world scenarios. For some tasks, it might not even be clear what the relevant state variables are, e.g. when manipulating deformable objects or articulated objects with unknown kinematic structures. In Chapter 6, we addressed this challenge by representing the state as a belief over object poses, which can be inferred from contact interactions. However, this approach still relies on a known object model and the true underlying state is captured by a pose in SE(3). Moving beyond known object models and full-state information requires methods that can learn compact, task-relevant state representations directly from raw sensory inputs, such as images or point clouds. The choice of state representation has a profound impact on the performance of planning and control algorithms, as it determines the complexity of the dynamics model, the expressiveness of the policy, and the efficiency of the optimisation process. Learning such representations is a challenging problem, as it requires disentangling relevant features from high-dimensional sensory data, while also ensuring that the learned representation is suitable for control (Ai et al., 2025). We consider this an important direction for future research, as it would significantly broaden the applicability of the methods presented in this thesis to more complex and unstructured environments. An orthogonal direction is to explore the use of *teacher-student distillation* (Miki et al., 2022; Chou and Tedrake, 2023; Yamada et al., 2024) to distill vision-based policies from state-based policies. This could possibly equally enable the use of the methods presented in this thesis in scenarios where only raw sensory inputs are available at test time, while still leveraging the benefits of state-based planning and control during training.

Distributional Robustness A core assumption throughout this thesis is that the uncertainty distributions are accurately represented by the available samples. While we account for the approximation error due to finite sampling, we do not consider

the possibility that the samples may not accurately reflect the true underlying distributions. This could be partially addressed by learning dynamics models that are able to better capture unmodelled dynamics, as discussed above. However, it would also be interesting to explore distributionally robust approaches that explicitly account for model misspecification, i.e. epistemic uncertainty, and distributional shifts (Chaouach et al., 2022; Hakobyan and Yang, 2021).

Beyond Unimodal Search Distributions Both VP-STO and CC-VPSTO rely on CMA-ES as the underlying optimisation algorithm, which uses a uni-modal Gaussian search distribution. While this has shown to be effective in practice, it also limits the expressiveness of the trajectory proposals and, consequently, the diversity of solutions that could be explored during optimisation. We have gone towards multi-modality with learned proposal distributions through flow-matching in GPC, but the data was still generated with a uni-modal Gaussian search distribution. While Sacks and Boots (2023) and Power and Berenson (2024) have recently explored learning more expressive search distributions specifically for MPPI, it would be interesting to explore multi-modal search distributions in VP-STO and CC-VPSTO, e.g. through multi-population evolution strategies instead of CMA-ES. Orthogonal to the expressiveness of the search distribution, but with the same goal of improving the efficiency of the sampling process, it would also be interesting to include better exploration strategies, e.g. through diversity measures from the reinforcement learning literature (Zahavy et al., 2022; Braun et al., 2025), into the sampling process.

Bibliography

Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. (2018). Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*.

Agboh, W. C. and Dogar, M. R. (2018). Real-time online re-planning for grasping under clutter and uncertainty. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 1–8. IEEE.

Ai, B., Tian, S., Shi, H., Wang, Y., Pfaff, T., Tan, C., Christensen, H. I., Su, H., Wu, J., and Li, Y. (2025). A review of learning-based dynamics models for robotic manipulation. *Science Robotics*, 10(106):eadt1497.

Ai, B., Tian, S., Shi, H., Wang, Y., Tan, C., Li, Y., and Wu, J. (2024). Robopack: Learning tactile-informed dynamics models for dense packing. In *Robotics: Science and Systems (RSS), 2024*.

Albu-Schäffer, A., Ott, C., and Hirzinger, G. (2007). A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *The international journal of robotics research*, 26(1):23–39.

Alvarez-Padilla, J., Zhang, J. Z., Kwok, S., Dolan, J. M., and Manchester, Z. (2025). Real-time whole-body control of legged robots with model-predictive path integral control. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14721–14727. IEEE.

Ames, A. D., Xu, X., Grizzle, J. W., and Tabuada, P. (2016). Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876.

Andrychowicz, O. M., Baker, B., Chociej, M., Józefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., Schneider, J., Sidor, S., Tobin, J., Welinder, P., Weng, L., and Zaremba, W. (2020). Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20.

Anitescu, M. and Potra, F. A. (1997). Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics*, 14(3):231–247.

Ankile, L., Jiang, Z., Duan, R., Shi, G., Abbeel, P., and Nagabandi, A. (2025). Residual off-policy rl for finetuning behavior cloning policies. *arXiv preprint arXiv:2509.19301*.

Arruda, E., Mathew, M. J., Kopicki, M., Mistry, M., Azad, M., and Wyatt, J. L. (2017). Uncertainty averse pushing with model predictive path integral control. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 497–502. IEEE.

Åström, K. J. (1965). Optimal control of markov processes with incomplete state information i. *Journal of mathematical analysis and applications*, 10:174–205.

Audet, C. and Kokkolaras, M. (2016). Blackbox and derivative-free optimization: theory, algorithms and applications. *Optimization and Engineering*, 17(1):1–2.

Bain, M. and Sammut, C. (1995). A framework for behavioural cloning. In *Machine intelligence 15*, pages 103–129.

Bajcsy, A., Bansal, S., Bronstein, E., Tolani, V., and Tomlin, C. J. (2019). An efficient reachability-based framework for provably safe autonomous navigation in unknown environments. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 1758–1765. IEEE.

Baltussen, T., Orrico, C., Katriniok, A., Heemels, W., and Krishnamoorthy, D. (2025). Value function approximation for nonlinear mpc: Learning a terminal cost function with a descent property. *arXiv preprint arXiv:2508.05804*.

Banker, T., Lawrence, N. P., and Mesbah, A. (2025). Local-global learning of interpretable control policies: The interface between mpc and reinforcement learning. *arXiv preprint arXiv:2503.13289*.

Belvedere, T., Cognetti, M., Oriolo, G., and Giordano, P. R. (2025). Sensitivity-aware model predictive control for robots with parametric uncertainty. *IEEE Transactions on Robotics*.

Ben-Tal, A. and Nemirovski, A. (1998). Robust convex optimization. *Mathematics of operations research*, 23(4):769–805.

Bertsekas, D. (2012). *Dynamic programming and optimal control: Volume I*, volume 4. Athena scientific.

Bertsekas, D. (2019). *Reinforcement learning and optimal control*, volume 1. Athena Scientific.

Bertsekas, D. and Shreve, S. E. (1996). *Stochastic optimal control: the discrete-time case*, volume 5. Athena Scientific.

Bhardwaj, M., Sundaralingam, B., Mousavian, A., Ratliff, N. D., Fox, D., Ramos, F., and Boots, B. (2022). Storm: An integrated framework for fast joint-space model-predictive control for reactive manipulation. In *Conference on Robot Learning*, pages 750–759. PMLR.

Bianchini, B., Halm, M., and Posa, M. (2023). Simultaneous learning of contact and continuous dynamics. In *Conference on Robot Learning*, pages 3966–3978.

PMLR.

Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al. (2024). π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*.

Blackmore, L., Ono, M., and Williams, B. C. (2011). Chance-constrained optimal path planning with obstacles. *IEEE Transactions on Robotics*, 27(6):1080–1094.

Boyd, S. P. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Branke, J., Kaußler, T., Smidt, C., and Schmeck, H. (2000). A multi-population approach to dynamic optimization problems. In *Evolutionary Design and Manufacture: Selected Papers from ACDM'00*, pages 299–307. Springer.

Braun, C. V., Auddy, S., and Toussaint, M. (2025). Trajectory first: A curriculum for discovering diverse policies. *arXiv preprint arXiv:2506.01568*.

Brown, L. D., Cai, T. T., and DasGupta, A. (2001). Interval estimation for a binomial proportion. *Statistical science*, 16(2):101–133.

Brudermüller, L., Berger, G. O., Jankowski, J., Bhattacharyya, R., Calinon, S., Jungers, R. M., and Hawes, N. (2025a). CC-VPSTO: Chance-Constrained Via-Point-Based Stochastic Trajectory Optimisation for Online Robot Motion Planning under Uncertainty. Manuscript under review at *The International Journal of Robotics Research*.

Brudermüller, L., Hung, B., Zhu, X., Wang, J., Hawes, N., Culbertson, P., and Le Cleac'h, S. (2025b). Generative models from and for sampling-based mpc: A bootstrapped approach for adaptive contact-rich manipulation. Manuscript under review at *IEEE Robotics and Automation Letters*.

Brudermüller, L., Jankowski, J., Toussaint, M., and Hawes, N. (2025c). Touch-based object localisation with spatially-aware belief entropy estimation. Manuscript under review at *IEEE International Conference on Robotics and Automation (ICRA)*.

Calafiore, G. C. and Campi, M. C. (2006). The scenario approach to robust control design. *IEEE Transactions on automatic control*, 51(5):742–753.

Chaouach, L. M., Boskos, D., and Oomen, T. (2022). Uncertain uncertainty in data-driven stochastic optimization: towards structured ambiguity sets. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 4776–4781. IEEE.

Chen, X. and Li, Y. (2023). An overview of differentiable particle filters for data-adaptive sequential bayesian inference. *arXiv preprint arXiv:2302.09639*.

Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., and Song, S. (2023). Diffusion policy: Visuomotor policy learning via action diffusion.

The International Journal of Robotics Research, page 02783649241273668.

Chi, C., Xu, Z., Pan, C., Cousineau, E., Burchfiel, B., Feng, S., Tedrake, R., and Song, S. (2024). Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*.

Chou, G. and Tedrake, R. (2023). Synthesizing stable reduced-order visuomotor policies for nonlinear systems via sums-of-squares optimization. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 624–631. IEEE.

Clark, A. (2021). Control barrier functions for stochastic systems. *Automatica*, 130:109688.

Conn, A. R., Scheinberg, K., and Vicente, L. N. (2009). *Introduction to derivative-free optimization*. SIAM.

Coumans, E. (2015). Bullet physics simulation. In *ACM SIGGRAPH 2015 Courses*, page 1.

Curtis, A., Kaelbling, L., and Jain, S. (2022). Task-directed exploration in continuous pomdps for robotic manipulation of articulated objects. *arXiv preprint arXiv:2212.04554*.

De Boor, C. and De Boor, C. (1978). *A practical guide to splines*, volume 27. springer New York.

de Groot, O., Ferranti, L., Gavrila, D. M., and Alonso-Mora, J. (2025). Scenario-based motion planning with bounded probability of collision. *The International Journal of Robotics Research*, 44(9):1507–1525.

DeMoss, B., Duckworth, P., Hawes, N., and Posner, I. (2023). Ditto: Offline imitation learning with world models. *arXiv preprint arXiv:2302.03086*.

Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794.

Du Toit, N. E. and Burdick, J. W. (2010). Robotic motion planning in dynamic, cluttered, uncertain environments. In *2010 IEEE International Conference on Robotics and Automation*, pages 966–973. IEEE.

Emmer, S., Kratz, M., and Tasche, D. (2013). What is the best risk measure in practice? a comparison of standard measures. *arXiv preprint arXiv:1312.1645*.

Erdmann, M. A. and Mason, M. T. (2002). An exploration of sensorless manipulation. *IEEE Journal on Robotics and Automation*, 4(4):369–379.

Erez, T. and Smart, W. D. (2012). A scalable method for solving high-dimensional continuous pomdps using local approximation. *arXiv preprint arXiv:1203.3477*.

Farshidian, F., Jelavic, E., Satapathy, A., Giftthaler, M., and Buchli, J. (2017). Real-time motion planning of legged robots: A model predictive control ap-

proach. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 577–584. IEEE.

Featherstone, R. (2008). *Rigid body dynamics algorithms*. Springer.

Feng, R., Yu, C., Deng, W., Hu, P., and Wu, T. (2025). On the guidance of flow matching. *arXiv preprint arXiv:2502.02150*.

Finn, C., Levine, S., and Abbeel, P. (2016). Guided cost learning: Deep inverse optimal control via policy optimization. In *International conference on machine learning*, pages 49–58. PMLR.

Fischer, J. and Tas, Ö. S. (2020). Information particle filter tree: An online algorithm for pomdps with belief-based rewards on continuous domains. In *International Conference on Machine Learning*, pages 3177–3187. PMLR.

Florence, P., Lynch, C., Zeng, A., Ramirez, O. A., Wahid, A., Downs, L., Wong, A., Lee, J., Mordatch, I., and Tompson, J. (2022). Implicit behavioral cloning. In *Conference on robot learning*, pages 158–168. PMLR.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (1995). *Bayesian data analysis*. Chapman and Hall/CRC.

Giftthaler, M., Neunert, M., Stäuble, M., Buchli, J., and Diehl, M. (2018). A family of iterative gauss-newton shooting methods for nonlinear optimal control. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9. IEEE.

GX-Chen, A., Prakash, J., Guo, J., Fergus, R., and Ranganath, R. (2025). Kl-regularized reinforcement learning is designed to mode collapse. *arXiv preprint arXiv:2510.20817*.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. (2018). Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. (2019). Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR.

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. (2025). Mastering diverse control tasks through world models. *Nature*, pages 1–7.

Hakobyan, A. and Yang, I. (2021). Wasserstein distributionally robust motion control for collision avoidance using conditional value-at-risk. *IEEE Transactions on Robotics*, 38(2):939–957.

Handa, A., Allshire, A., Makoviychuk, V., Petrenko, A., Singh, R., Liu, J., Makoviichuk, D., Van Wyk, K., Zhurkevich, A., Sundaralingam, B., et al. (2023). Dextreme: Transfer of agile in-hand manipulation from simulation to reality. In

2023 IEEE International Conference on Robotics and Automation (ICRA), pages 5977–5984. IEEE.

Hansen, N. (2016). The CMA Evolution Strategy: A Tutorial.

Hansen, N. and Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195.

Hansen, N., Su, H., and Wang, X. (2023). Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828*.

Hansen, N. A., Su, H., and Wang, X. (2022). Temporal difference learning for model predictive control. In *International Conference on Machine Learning*, pages 8387–8406. PMLR.

Hargraves, C. R. and Paris, S. W. (1987). Direct trajectory optimization using nonlinear programming and collocation. *Journal of guidance, control, and dynamics*, 10(4):338–342.

Hauser, K. (2011). Randomized belief-space replanning in partially-observable continuous spaces. In *Algorithmic Foundations of Robotics IX: Selected Contributions of the Ninth International Workshop on the Algorithmic Foundations of Robotics*, pages 193–209. Springer.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Hogan, F. R. and Rodriguez, A. (2020). Reactive planar non-prehensile manipulation with hybrid model predictive control. *The International Journal of Robotics Research*, 39(7):755–773.

Hogan, N. (1984). Impedance control: An approach to manipulation. In *1984 American control conference*, pages 304–313. IEEE.

Holland, J. H. (1992). Genetic algorithms. *Scientific american*, 267(1):66–73.

Horowitz, M. and Burdick, J. (2013). Interactive non-prehensile manipulation for grasping via pomdps. In *2013 IEEE International Conference on Robotics and Automation*, pages 3257–3264. IEEE.

Howell, T., Gileadi, N., Tunyasuvunakool, S., Zakka, K., Erez, T., and Tassa, Y. (2022a). Predictive sampling: Real-time behaviour synthesis with mujoco. *arXiv preprint arXiv:2212.00541*.

Howell, T. A., Cleac'h, S. L., Brüdigam, J., Kolter, J. Z., Schwager, M., and Manchester, Z. (2022b). Dojo: A differentiable physics engine for robotics. *arXiv preprint arXiv:2203.00806*.

Hu, H. and Fisac, J. F. (2022). Active uncertainty reduction for human-robot interaction: An implicit dual control approach. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 385–401. Springer.

Huang, S., Chen, Q., Zhang, X., Sun, J., and Schwager, M. (2025). Particleformer: A 3d point cloud world model for multi-object, multi-material robotic manipulation. *arXiv preprint arXiv:2506.23126*.

Ichter, B., Harrison, J., and Pavone, M. (2018). Learning sampling distributions for robot motion planning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7087–7094. IEEE.

Jain, A. K., Mohta, V., Kim, S., Bhardwaj, A., Ren, J., Feng, Y., Choudhury, S., and Swamy, G. (2025). A smooth sea never made a skilled `sailor`: Robust imitation via learning to search. *arXiv preprint arXiv:2506.05294*.

Jankowski*, J., Brudermüller*, L., Hawes, N., and Calinon, S. (2023). VP-STO: Via-point-based stochastic trajectory optimization for reactive robot behavior. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10125–10131.

Jankowski, J., Brudermüller, L., Hawes, N., and Calinon, S. (2025a). Robust pushing: Exploiting quasi-static belief dynamics and contact-informed optimization. *The International Journal of Robotics Research*, page 02783649251318046.

Jankowski, J., Klink, P., Posner, I., Gundogdu, E., Park, K., and Erdogan, C. (2025b). Guaranteed SE(3)-equivariant control via hand-centric behavior cloning. In *Workshop on Generalizable Priors for Robot Manipulation at CoRL 2025*.

Jankowski, J., Racca, M., and Calinon, S. (2022). From key positions to optimal basis functions for probabilistic adaptive control. *IEEE Robotics and Automation Letters*, 7(2):3242–3249.

Jankowski, J. M. (2025). *A Stochastic Approach to Contact-rich Manipulation*. doctoral thesis, École Polytechnique Fédérale de Lausanne, Lausanne.

Jawale, N., Boots, B., Sundaralingam, B., and Bhardwaj, M. (2025). Dynamic non-prehensile object transport via model-predictive reinforcement learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3647–3653. IEEE.

Johannink, T., Bahl, S., Nair, A., Luo, J., Kumar, A., Loskyll, M., Ojea, J. A., Solowjow, E., and Levine, S. (2019). Residual reinforcement learning for robot control. In *2019 international conference on robotics and automation (ICRA)*, pages 6023–6029. IEEE.

Jordana, A., Zhang, J., Amigo, J., and Righetti, L. (2025). An introduction to zero-order optimization techniques for robotics. *arXiv preprint arXiv:2506.22087*.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting

in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134.

Kaelbling, L. P. and Lozano-Pérez, T. (2013). Integrated task and motion planning in belief space. *The International Journal of Robotics Research*, 32(9-10):1194–1227.

Kingma, D. P., Welling, M., et al. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392.

Koval, M. C., Pollard, N. S., and Srinivasa, S. S. (2016). Pre-and post-contact policy decomposition for planar contact manipulation under uncertainty. *The International Journal of Robotics Research*, 35(1-3):244–264.

Kurtz, V. and Burdick, J. W. (2025). Generative predictive control: Flow matching policies for dynamic and difficult-to-demonstrate tasks. *arXiv preprint arXiv:2502.13406*.

Le Cleac'h, S., Howell, T. A., Yang, S., Lee, C.-Y., Zhang, J., Bishop, A., Schwager, M., and Manchester, Z. (2024). Fast contact-implicit model predictive control. *IEEE Transactions on Robotics*, 40:1617–1629.

Lew, T., Bonalli, R., and Pavone, M. (2023). Risk-averse trajectory optimization via sample average approximation. *IEEE Robotics and Automation Letters*, 9(2):1500–1507.

Lew, T., Greiff, M., Djeumou, F., Suminaka, M., Thompson, M., and Subosits, J. (2025). Risk-averse model predictive control for racing in adverse conditions. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8508–8514. IEEE.

Li, C., Krause, A., and Hutter, M. (2025). Robotic world model: A neural network simulator for robust policy optimization in robotics. *arXiv preprint arXiv:2501.10100*.

Li, W. and Todorov, E. (2004). Iterative linear quadratic regulator design for nonlinear biological movement systems. In *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO,*, pages 222–229. INSTICC, SciTePress.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. (2022). Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*.

Lipman, Y., Havasi, M., Holderrieth, P., Shaul, N., Le, M., Karrer, B., Chen, R. T., Lopez-Paz, D., Ben-Hamu, H., and Gat, I. (2024). Flow matching guide and code. *arXiv preprint arXiv:2412.06264*.

Liu, R., Hou, A., Li, S., and Yin, X. (2025). Vh-diffuser: Variable horizon diffusion planner for time-aware goal-conditioned trajectory planning. *arXiv preprint arXiv:2509.11930*.

Lynch, K. M. and Park, F. C. (2017). *Modern robotics*. Cambridge University Press.

Majumdar, A. and Pavone, M. (2017). How Should a Robot Assess Risk? Towards an Axiomatic Theory of Risk in Robotics.

Majumdar, A. and Tedrake, R. (2017). Funnel libraries for real-time robust feedback motion planning. *The International Journal of Robotics Research*, 36(8):947–982.

Manchester, Z. and Kuindersma, S. (2019). Variational contact-implicit trajectory optimization. In *Robotics Research: The 18th International Symposium ISRR*, pages 985–1000. Springer.

Marques, J. M. C., Dengler, N., Zaenker, T., Mucke, J., Wang, S., Bennewitz, M., and Hauser, K. (2025). Map space belief prediction for manipulation-enhanced mapping. In *Robotics: Science and Systems (RSS), 2025*.

Mason, M. T. (1986). Mechanics and planning of manipulator pushing operations. *The International Journal of Robotics Research*, 5(3):53–71.

Mason, M. T. (2001). *Mechanics of robotic manipulation*. MIT press.

Matyas, J. et al. (1965). Random optimization. *Automation and Remote control*, 26(2):246–253.

Mayne, D. Q. (1973). Differential dynamic programming–a unified approach to the optimization of dynamic systems. In *Control and dynamic systems*, volume 10, pages 179–254. Elsevier.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814.

Melon, O., Geisert, M., Surovik, D., Havoutis, I., and Fallon, M. (2020). Reliable trajectories for dynamic quadrupeds using analytical costs and learned initializations. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1410–1416. IEEE.

Mesbah, A. (2016). Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, 36(6):30–44.

Mesbah, A., Streif, S., Findeisen, R., and Braatz, R. D. (2014). Stochastic nonlinear model predictive control with probabilistic constraints. In *2014 American control conference*, pages 2413–2419. IEEE.

Mesbah, A., Wabersich, K. P., Schoellig, A. P., Zeilinger, M. N., Lucia, S., Badgwell, T. A., and Paulson, J. A. (2022). Fusion of machine learning and mpc under uncertainty: What advances are on the horizon? In *2022 American Control Conference (ACC)*, pages 342–357. IEEE.

Mestres, P., Werner, B., Cosner, R. K., and Ames, A. D. (2025). Probabilistic control barrier functions: Safety in probability for discrete-time stochastic systems. *arXiv preprint arXiv:2510.01501*.

Miki, T., Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V., and Hutter, M. (2022). Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science robotics*, 7(62):eabk2822.

Mohamed, I. S., Ali, M., and Liu, L. (2025). Chance-constrained sampling-based mpc for collision avoidance in uncertain dynamic environments. *IEEE Robotics and Automation Letters*.

Moss, R. J., Corso, A., Caers, J., and Kochenderfer, M. J. (2023). Betazero: Belief-state planning for long-horizon pomdps using learned approximations. *arXiv preprint arXiv:2306.00249*.

Moss, R. J., Jamgochian, A., Fischer, J., Corso, A., and Kochenderfer, M. (2024). Chance-constrained pomdp planning with learned neural network surrogates. In *IJCAI 2024 Workshop on Trustworthy Interactive Decision-Making with Foundation Models*.

Muratore, F., Ramos, F., Turk, G., Yu, W., Gienger, M., and Peters, J. (2022). Robot learning from randomized simulations: A review. *Frontiers in Robotics and AI*, 9:799893.

Nakkiran, P., Bradley, A., Zhou, H., and Advani, M. (2024). Step-by-step diffusion: An elementary tutorial. *arXiv preprint arXiv:2406.08929*.

Nematollahi, I., DeMoss, B., Chandra, A. L., Hawes, N., Burgard, W., and Posner, I. (2025). Lumos: Language-conditioned imitation learning with world models. *arXiv preprint arXiv:2503.10370*.

Pang, T. and Tedrake, R. (2021). A convex quasistatic time-stepping scheme for rigid multibody systems with contact and friction. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6614–6620. IEEE.

Papadimitriou, C. H. and Tsitsiklis, J. N. (1987). The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450.

Pezzato, C., Salmi, C., Trevisan, E., Spahn, M., Alonso-Mora, J., and Corbato, C. H. (2025). Sampling-based model predictive control leveraging parallelizable physics simulations. *IEEE Robotics and Automation Letters*.

Platt, R., Kaelbling, L., Lozano-Perez, T., and Tedrake, R. (2011). Simultaneous localization and grasping as a belief space control problem. In *International Symposium on Robotics Research*, volume 2.

Platt, R., Tedrake, R., Kaelbling, L. P., and Lozano-Perez, T. (2010). Belief space planning assuming maximum likelihood observations. In *Robotics: Science and Systems*.

Posa, M. and Tedrake, R. (2013). Direct trajectory optimization of rigid body dynamical systems through contact. In *Algorithmic Foundations of Robotics X: Proceedings of the Tenth Workshop on the Algorithmic Foundations of Robotics*,

pages 527–542. Springer.

Powell, M. J. (1987). Radial basis functions for multivariable interpolation: a review. *Algorithms for approximation*, pages 143–167.

Powell, M. J. (1994). A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis*, pages 51–67. Springer.

Power, T. and Berenson, D. (2024). Learning a generalizable trajectory sampling distribution for model predictive control. *IEEE Transactions on Robotics*, 40:2111–2127.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Rawlings, J. B., Mayne, D. Q., Diehl, M., et al. (2020). *Model predictive control: theory, computation, and design*, volume 2. Nob Hill Publishing Madison, WI.

Rigter, M., Lacerda, B., and Hawes, N. (2022). Rambo-rl: Robust adversarial model-based offline reinforcement learning. *Advances in neural information processing systems*, 35:16082–16097.

Ros, R. and Hansen, N. (2008). A simple modification in cma-es achieving linear time and space complexity. In *International conference on parallel problem solving from nature*, pages 296–305. Springer.

Ross, S. and Bagnell, D. (2010). Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668. JMLR Workshop and Conference Proceedings.

Ross, S., Gordon, G., and Bagnell, D. (2011). A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings.

Röstel, L., Sievers, L., Pitz, J., and Bäuml, B. (2022). Learning a state estimator for tactile in-hand manipulation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4749–4756. IEEE.

Rubinstein, R. (1999). The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1(2):127–190.

Sacks, J. and Boots, B. (2023). Learning sampling distributions for model predictive control. In *Conference on Robot Learning*, pages 1733–1742. PMLR.

Scholz, J. and Turner, R. E. (2025). Warm starts accelerate generative modelling. *arXiv preprint arXiv:2507.09212*.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. (2020). Mastering atari,

go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shekhar, R. C. (2012). *Variable horizon model predictive control: robustness and optimality*. PhD thesis.

Shetty, S., Xue, T., and Calinon, S. (2024). Generalized policy iteration using tensor approximation for hybrid control. In *The Twelfth International Conference on Learning Representations*.

Siciliano, B. and Villani, L. (1999). *Robot force control*. Springer Science & Business Media.

Silverman, B. W. (2018). *Density estimation for statistics and data analysis*. Routledge.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2020). Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Stewart, D. E. (2000). Rigid-body dynamics with friction and impact. *SIAM review*, 42(1):3–39.

Sullivan, T. J. (2015). *Introduction to uncertainty quantification*, volume 63. Springer.

Sun, J., Jiang, Y., Qiu, J., Nobel, P., Kochenderfer, M. J., and Schwager, M. (2023). Conformal prediction for uncertainty-aware planning with diffusion dynamics model. *Advances in Neural Information Processing Systems*, 36:80324–80337.

Sunberg, Z. N. and Kochenderfer, M. J. (2018). Online algorithms for pomdps with continuous state, action, and observation spaces. In *Twenty-Eighth International Conference on Automated Planning and Scheduling*.

Sutton, R. S., Barto, A. G., et al. (1998). *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Tamar, A., Thomas, G., Zhang, T., Levine, S., and Abbeel, P. (2017). Learning from the hindsight plan—episodic mpc improvement. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 336–343. IEEE.

Tassa, Y., Erez, T., and Todorov, E. (2012). Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913. IEEE.

Tedrake, R. (2023). *Underactuated Robotics*.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE.

Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE.

Torne, M., Tang, A., Liu, Y., and Finn, C. (2025). Learning long-context diffusion policies via past-token prediction. *arXiv preprint arXiv:2505.09561*.

Tremblay, J.-F., Meger, D., Hogan, F. R., and Dudek, G. (2025). Learning active tactile perception through belief-space control. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8702–8708. IEEE.

Trevisan, E., Mustafa, K. A., Notten, G., Wang, X., and Alonso-Mora, J. (2025). Dynamic risk-aware mppi for mobile robots in crowds via efficient monte carlo approximations. *arXiv preprint arXiv:2506.21205*.

Van Den Oord, A., Vinyals, O., et al. (2017). Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Wang, C., Meng, Y., Smith, S. L., and Liu, J. (2021a). Safety-critical control of stochastic systems using stochastic control barrier functions. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 5924–5931. IEEE.

Wang, D., Hu, B., Song, S., Walters, R., and Platt, R. (2025a). A practical guide for incorporating symmetry in diffusion policy. *arXiv preprint arXiv:2505.13431*.

Wang, Y., Guo, H., Wang, S., Qian, L., and Lan, X. (2025b). Bootstrapped model predictive control. *arXiv preprint arXiv:2503.18871*.

Wang, Z., So, O., Gibson, J., Vlahov, B., Gandhi, M. S., Liu, G.-H., and Theodorou, E. A. (2021b). Variational inference mpc using tsallis divergence. *arXiv preprint arXiv:2104.00241*.

Wensing, P. M., Posa, M., Hu, Y., Escande, A., Mansard, N., and Del Prete, A. (2023). Optimization-based control for dynamic legged robots. *IEEE Transactions on Robotics*, 40:43–63.

Williams, G., Aldrich, A., and Theodorou, E. A. (2017). Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 40(2):344–357.

Wu, P., Escontrela, A., Hafner, D., Abbeel, P., and Goldberg, K. (2023). Daydreamer: World models for physical robot learning. In *Conference on robot learning*, pages 2226–2240. PMLR.

Xu, Z., Wu, J., Zeng, A., Tenenbaum, J. B., and Song, S. (2019). Densephysnet: Learning dense physical object representations via multi-step dynamic interactions. In *Robotics: Science and Systems (RSS), 2019*.

Xue, H., Pan, C., Yi, Z., Qu, G., and Shi, G. (2025). Full-order sampling-based mpc for torque-level locomotion control via diffusion-style annealing. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4974–4981. IEEE.

Yamada, J., Murali, A., Mandlekar, A., Eppner, C., Posner, I., and Sundaralingam, B. (2025a). Grasp-mpc: Closed-loop visual grasping via value-guided model predictive control. *arXiv preprint arXiv:2509.06201*.

Yamada, J., Rigter, M., Collins, J., and Posner, I. (2024). Twist: Teacher-student world model distillation for efficient sim-to-real transfer. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9190–9196. IEEE.

Yamada, J., Zhong, S., Collins, J., and Posner, I. (2025b). D-cubed: Latent diffusion trajectory optimisation for dexterous deformable manipulation. In *Conference on Robot Learning (CoRL)*.

Yin, J., So, O., Yu, E. Y., Fan, C., and Tsiotras, P. (2025). Safe beyond the horizon: Efficient sampling-based mpc with neural control barrier functions. *arXiv preprint arXiv:2502.15006*.

Yin, J., Zhang, Z., and Tsiotras, P. (2022). Risk-aware model predictive path integral control using conditional value-at-risk. *arXiv preprint arXiv:2209.12842*.

Yu, K.-T., Bauza, M., Fazeli, N., and Rodriguez, A. (2016). More than a million ways to be pushed. a high-fidelity experimental dataset of planar pushing. In *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 30–37. IEEE.

Yuan, X., Mu, T., Tao, S., Fang, Y., Zhang, M., and Su, H. (2024). Policy decorator: Model-agnostic online refinement for large policy model. *arXiv preprint arXiv:2412.13630*.

Zahavy, T., Schroecker, Y., Behbahani, F., Baumli, K., Flennerhag, S., Hou, S., and Singh, S. (2022). Discovering policies with domino: Diversity optimization maintaining near optimality. *arXiv preprint arXiv:2205.13521*.

Zhang, F. and Gienger, M. (2024). Affordance-based robot manipulation with flow matching. *arXiv preprint arXiv:2409.01083*.

Zhang, J. Z., Howell, T. A., Yi, Z., Pan, C., Shi, G., Qu, G., Erez, T., Tassa, Y., and Manchester, Z. (2025). Whole-body model-predictive control of legged robots with mujoco. *arXiv preprint arXiv:2503.04613*.

Zhao, T. Z., Kumar, V., Levine, S., and Finn, C. (2023). Learning fine-grained

bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*.

Zhao, W., Queralta, J. P., and Westerlund, T. (2020). Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)*, pages 737–744. IEEE.

Zhong, M., Johnson, M., Tassa, Y., Erez, T., and Todorov, E. (2013). Value function approximation and model predictive control. In *2013 IEEE symposium on adaptive dynamic programming and reinforcement learning (ADPRL)*, pages 100–107. IEEE.

Zhou, G., Pan, H., LeCun, Y., and Pinto, L. (2024). Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*.

Zhou, K. and Doyle, J. C. (1998). *Essentials of robust control*, volume 104. Prentice hall Upper Saddle River, NJ.

Zhu, X., Chen, Y., Sun, L., Niroui, F., Cleac'h, S. L., Wang, J., and Fang, K. (2025). Versatile loco-manipulation through flexible interlimb coordination. *arXiv preprint arXiv:2506.07876*.

A

Appendices of Chapter 4

Generative Models From and For Sampling-Based MPC: A Bootstrapped Approach For Adaptive Contact-Rich Manipulation

Lara Brudermüller[1,2,*], Brandon Hung[2], Xinghao Zhu[2], Jiuguang Wang[2], Nick Hawes[1], Preston Culbertson[2,3,†], Simon Le Cleac'h[2,†]

***Abstract*— We present a generative predictive control (GPC) framework that amortizes sampling-based Model Predictive Control (SPC) by bootstrapping it with conditional flow-matching models trained on SPC control sequences collected in simulation. Unlike prior work relying on iterative refinement or gradient-based solvers, we show that meaningful proposal distributions can be learned directly from noisy SPC data, enabling more efficient and informed sampling during online planning. We further demonstrate, for the first time, the application of this approach to real-world contact-rich loco-manipulation with a quadruped robot. Extensive experiments in simulation and on hardware show that our method improves sample efficiency, reduces planning horizon requirements, and generalizes robustly across task variations.**

I. Introduction

Reactive contact-rich (loco-)manipulation in high-dimensional state and action spaces poses significant challenges for real-time control. Sampling-based Model Predictive Control, or sampling-based predictive control (SPC), offers a principled framework to address these challenges by solving trajectory optimization problems online with a model in the loop, enabling adaptive behavior and constraint satisfaction [1]–[4]. However, the computational cost of forward simulation, combined with the challenge of effectively exploring the search space in high-dimensional, contact-rich environments, limits the applicability of real-time optimization to more complex behaviors and higher-frequency control.

To address this, a promising line of work seeks to amortize the computational burden of online optimization by shifting it to an offline phase [5]–[7]. The key idea is to collect data, either from expert demonstrations or from model-based planning, and use it to train a generative model that captures the distribution of useful actions or control sequences. At test time, this model can then be used to guide or warmstart the sampling distribution in MPC, drastically improving efficiency and solution quality by focusing sampling on high-likelihood, constraint-satisfying regions of the action space.

Recent advances in generative modeling, particularly diffusion and flow-matching models, have shown strong performance in learning expressive policies for dexterous manipulation tasks [8]–[10]. Similarly, offline model-based reinforcement learning methods [11], [12] leverage precomputed data to enable fast runtime control via policy networks trained to approximate optimal solutions. Yet, these methods are often limited by the scope of their training data and struggle to generalize to out-of-distribution (OOD) states or tasks. In response to these limitations, several recent works demonstrate that bootstrapping online planners with offline-trained generative models leads to faster convergence, better exploration, and more robust performance in complex environments [7], [13]–[15]. In this paper, we adopt this amortized optimization perspective, focusing on how offline data collection and generative modeling can be used to accelerate and guide online sampling-based MPC in contact-rich, high-dimensional settings while maintaining the flexibility and adaptability of online optimization.

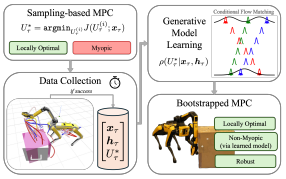


Fig. 1. *Generative predictive control (GPC) framework for bootstrapping sampling-based Model Predictive Control (SPC).* We collect open-loop control sequences from an SPC algorithm in simulation and use them to train a generative proposal distribution. At test time, this model guides and amortizes online MPC, enabling non-myopic, constraint-satisfying behavior with improved sample efficiency and robustness in contact-rich, high-dimensional settings.

Contributions: We propose a *generative predictive control (GPC) framework* that *bootstraps* SPC with conditional flow-matching models trained on SPC control sequences collected in simulation. To the best of our knowledge, we are the first to show that meaningful *proposal distributions can be learned directly from noisy SPC data* without requiring expert refinement or numerical solvers. We are also the first to demonstrate this approach on *real hardware in a contact-rich loco-manipulation task*, and show that it improves sample efficiency and *generalizes robustly to task variations* in both simulation and real world.

[1]Oxford Robotics Institute, University of Oxford, UK.
[2]Robotics and AI Institute (RAI), Boston, USA.
[3]Cornell University, USA.
*This work was done while Lara Brudermüller was an intern at the Robotics and AI Institute.
†Preston Culbertson and Simon Le Cleac'h advised this work equally.

APPENDIX

A. Implementation Details

We describe the implementation of our generative model and sampling-based MPC algorithm, used for both offline data collection and online control.

Data Collection: We collect training data using a sampling-based MPC controller in simulation for a maximum of 2500 time steps per episodes in both tasks, which generates open-loop control sequences. For Push-T, trajectories are cubic splines with 4 control points; for Spot, they consist of 4 linearly interpolated waypoints. All data is represented in the world frame, not relative to the robot. Although we tested robot-centric representations, they did not yield performance improvements. We collected 67,667 Push-T sequences from 1,000 successful episodes and 211,832 Spot sequences from 1,700 episodes. For evaluation, we use fixed sets of randomly sampled initial states for each task, shared across all methods and runs to ensure consistency.

Training Details: We implemented our baseline CVAE and conditional flow-matching models as MLPs trained with a batch size of 40,000 and the Adam optimizer with a learning rate 0.0001, cosine annealing schedule, 500 warmup steps over 1,000 epochs. The model predicts 4 control points conditioned on the current robot and object state and the previous replanning state (history length = 1). Orientations are represented using sine-cosine encodings of yaw angles. Although Spot expects velocity commands, we predict absolute positions and convert them to velocities via finite differences during online control.

Cost Functions: Both tasks use a weighted sum of cost terms computed over the full-resolution control sequence (0.01s for Push-T, 0.02s for Spot). The cost components are:

1) *Robot-Object Proximity:* L2 distance between the robot and object. For Spot, we penalize both torso and end-effector distances.
2) *Velocity Penalty:* L2 norm of robot joint velocities.
3) *Goal Reaching:* L2 distance and angle difference between object and goal, plus a progress term penalizing lack of advancement over time.
4) *Joint Limits (Spot):* Penalty for exceeding arm joint limits (leg joints are handled by the low-level policy).
5) *Fall Penalty (Spot):* Large penalty if the torso height drops below a threshold, preventing falls.
6) *Object Tipping (Spot):* Large penalty if the chair's z-axis deviates from vertical, discouraging it from tipping over.

Weights were tuned empirically and will be provided in the released code.

B

Appendices of Chapter 5

Journal Title
XX(X):1–22

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

CC-VPSTO: Chance-Constrained Via-Point-Based Stochastic Trajectory Optimisation for Online Robot Motion Planning under Uncertainty

Lara Brudermüller[1], Guillaume O. Berger[2], Julius Jankowski[3], Raunak Bhattacharyya[1], Sylvain Calinon[3], Raphaël M. Jungers[2], and Nick Hawes[1]

Abstract

Reliable robot autonomy hinges on decision-making systems that account for uncertainty without imposing overly conservative restrictions on the robot's action space. We introduce *Chance-Constrained Via-Point-Based Stochastic Trajectory Optimisation (CC-VPSTO)*, a real-time capable framework for generating task-efficient robot trajectories that satisfy constraints with high probability by formulating stochastic control as a chance-constrained optimisation problem. Since such problems are generally intractable, we propose a deterministic surrogate formulation based on Monte Carlo sampling, solved efficiently with gradient-free optimisation. To address bias in naïve sampling approaches, we quantify approximation error and introduce padding strategies to improve reliability. We focus on three challenges: (i) sample-efficient constraint approximation, (ii) conditions for surrogate solution validity, and (iii) online optimisation. Integrated into a receding-horizon MPC framework, CC-VPSTO enables enables reactive, task-efficient control under uncertainty, balancing constraint satisfaction and performance in a principled manner. The strengths of our approach lie in its generality, *i.e.*, no assumptions on the underlying uncertainty distribution, system dynamics, cost function, or the form of inequality constraints; and its applicability to online robot motion planning. We demonstrate the validity and efficiency of our approach in both simulation and on a Franka Emika robot. Videos and additional material are made available here.

Keywords

Chance-Constrained Optimisation, Stochastic Model Predictive Control, Trajectory Optimisation

1 Introduction

Uncertainty is inherent to most real-world robotics applications, arising from noisy sensors, imprecise actuators, and incomplete or evolving knowledge of the environment. Effectively managing this uncertainty is essential for achieving reliable and efficient robot behaviour, particularly in online motion planning tasks that require fast adaptation to new information. In this work, we adopt a *chance-constrained* perspective, where constraints such as collision avoidance (cf. Fig. 1), force limits, or task completion cannot be guaranteed but must instead be satisfied with high probability (Prékopa 2013; Dai et al. 2019). Unlike traditional robust control methods (Köhler et al. 2023; Majumdar and Tedrake 2017; Badings et al. 2023) that optimise for the worst-case scenario under *bounded uncertainty*, chance constraints enable a more general, *probabilistic* treatment of uncertainty (Margellos et al. 2014; Schildbach et al. 2014), allowing for more explicit trade-offs between constraint satisfaction and task efficiency.

Crucially, we are interested in an *online* robot motion planning setting where constraint violations are undesirable but not catastrophic, and where performance (*e.g.*, motion duration) remains important. Our objective is to trade off constraint satisfaction and task performance in a principled manner that avoids unnecessary conservatism. While Model Predictive Control (MPC) can implicitly provide some robustness via frequent replanning, it typically relies on deterministic models, leading to brittle, myopic behaviour in stochastic settings. Incorporating probabilistic information directly into the control loop remains a key challenge, but is essential for enabling more flexible and robust decision-making in uncertain environments.

Chance-constrained formulations, which require that constraints must be satisfied with high probability (*e.g.*, at least 95%), offer a natural solution but are in general intractable (Blackmore et al. 2010). One common strategy is to approximate the chance constraint and reformulate the problem as a deterministic surrogate that can be addressed using standard optimisation techniques. However, identifying a suitable approximation is often non-trivial. Common approaches either introduce significant conservatism at the cost of task efficiency (Lew et al. 2023; Calafiore and Campi 2006), or, like naïve sample

[1]Oxford Robotics Institute, University of Oxford, UK
[2]ICTEAM, UCLouvain, Belgium
[3]Idiap Research Institute & Ecole Polytechnique Fédérale de Lausanne (EPFL), CH

Corresponding author:
Lara Brudermüller, Oxford Robotics Institute,
23 Banbury Rd, Oxford OX2 6NN, UK
Email: larab@robots.ox.ac.uk

Prepared using sagej.cls *[Version: 2017/01/17 v1.20]*

APPENDIX

7 Trajectory Representation

The way we represent trajectories is based on previous work showing that the closed-form solution to the following optimisation problem

$$\begin{aligned} &\min \quad \int_0^1 \boldsymbol{q}''(s)^\top \boldsymbol{q}''(s) ds \\ \text{s.t.} \quad &\boldsymbol{q}(s_n) = \boldsymbol{q}_n, \quad n = 1, ..., N \\ &\boldsymbol{q}(0) = \boldsymbol{q}_0, \boldsymbol{q}'(0) = \boldsymbol{q}'_0,\ \boldsymbol{q}(1) = \boldsymbol{q}_T, \boldsymbol{q}'(1) = \boldsymbol{q}'_T \end{aligned} \tag{4}$$

is given by cubic splines (Zhang et al. 1997) and that it can be formulated as a weighted superposition of basis functions (Jankowski et al. 2022). Hence, the robot's configuration is defined as $\boldsymbol{q}(s) = \boldsymbol{\Phi}(s)\boldsymbol{w} \in \mathbb{R}^D$, with D being the number of degrees of freedom. The matrix $\boldsymbol{\Phi}(s)$ contains the basis functions which are weighted by the vector $\boldsymbol{w}$*. The trajectory is defined on the interval $\mathcal{S} = [0, 1]$, while the time t maps to the phase variable $s = \frac{t}{T} \in \mathcal{S}$ with T being the total duration of the trajectory. Consequently, joint velocities and accelerations along the trajectory are given by $\dot{\boldsymbol{q}}(s) = \frac{1}{T}\boldsymbol{\Phi}'(s)\boldsymbol{w}$ and $\ddot{\boldsymbol{q}}(s) = \frac{1}{T^2}\boldsymbol{\Phi}''(s)\boldsymbol{w}$, respectively†. The basis function weights $\boldsymbol{w}$ include the trajectory constraints consisting of the boundary condition parameters $\boldsymbol{w}_{\text{bc}} = [\boldsymbol{q}_0^\top, \boldsymbol{q}_0'^\top, \boldsymbol{q}_T^\top, \boldsymbol{q}_T'^\top]^\top$ and N via-points the trajectory has to pass through $\boldsymbol{q}_{\text{via}} = [\boldsymbol{q}_1^\top, \dots, \boldsymbol{q}_N^\top]^\top \in \mathbb{R}^{DN}$, such that $\boldsymbol{w} = [\boldsymbol{q}_{\text{via}}^\top, \boldsymbol{w}_{\text{bc}}^\top]^\top$. Throughout this paper, the via-point timings s_n are assumed to be uniformly distributed in $\mathcal{S}$. Note, that boundary velocities map to boundary derivatives w.r.t. s by multiplying them with the total duration T, *i.e.*, $\boldsymbol{q}'_0 = T\dot{\boldsymbol{q}}_0$ and $\boldsymbol{q}'_T = T\dot{\boldsymbol{q}}_T$. Furthermore, the optimisation problem in Eq. (4) minimizes not only the objective $\boldsymbol{q}''(s)$, but also the integral over accelerations, since $\boldsymbol{q}''(s) = T^2\ddot{\boldsymbol{q}}(s)$ and thus the objective $\int_0^1 \ddot{\boldsymbol{q}}(s)^\top \ddot{\boldsymbol{q}}(s) ds$ directly maps to $\frac{1}{T^4}\int_0^1 \boldsymbol{q}''(s)^\top \boldsymbol{q}''(s) ds$, corresponding to the control effort. It is minimal *iff* the objective in Eq. (4) is minimal. As a result, this trajectory representation provides a linear mapping from via points, boundary conditions and the movement duration to a time-continuous and smooth trajectory.

CC-VPSTO, analogously to VP-STO, exploits this explicit parameterisation with via-points and boundary conditions by optimizing *only the via-points* while keeping the predefined boundary condition parameters fixed.

8 Baseline for Offline Simulation Experiments: Derivation and Background

We present an alternative approach to approximate the chance constraint in Eq. (1) for the special case of obstacle collision avoidance, which we use as a baseline. For this, we leverage statistical learning theory (Shalev-Shwartz and Ben-David 2014; Mohri et al. 2018) to obtain an alternative confidence-based bound for k_{thresh}, which we call k_{rad}. This bound is more theoretically complete, as it does not require the independence of the Bernoulli variables, but we also note that it is more conservative, computationally expensive, and, less general since it is limited to a specific motion planning problem.

8.1 Preliminaries on Statistical Learning Theory

We remind the concept of *Rademacher complexity* which is a measure used in statistical learning theory to quantify the complexity of a class of functions with respect to a given dataset. The intuition is as follows: if the constraint function g is "simple" (*e.g.*, a constant or linear function), then the complexity of the class $\mathcal{F}$, defined later based on g, will be low. As established by Proposition 1 in Mohri et al. (2018), this implies that the "generalization property" of $\mathcal{F}$ is good. In our case, this means that if a solution $\boldsymbol{x}$ has a small rate of violating constraint g with respect to N i.i.d. samples, then there is a good chance that the actual probability of constraint violation of $\boldsymbol{x}$ is small too.

First, we introduce the notion of *Rademacher complexity* and the associated *generalization* result:

Definition 1. If $\mathcal{F}$ is a (possibly infinite) set of functions from a set $\mathcal{D}$ to $\mathbb{R}$, *i.e.*, $\mathcal{F} \subseteq \mathbb{R}^{\mathcal{D}}$, and $D = \{\boldsymbol{\delta}_1, \dots, \boldsymbol{\delta}_N\}_{i=1}^N$ is a set of N elements of $\mathcal{D}$, then the *Rademacher complexity of $\mathcal{F}$ with respect to D* is defined by

$$R_D(\mathcal{F}) = E_{\boldsymbol{\sigma}_1,\dots,\boldsymbol{\sigma}_N}\left[\sup_{f\in\mathcal{F}} \frac{1}{N}\sum_{i=1}^N \boldsymbol{\sigma}_i f(\boldsymbol{\delta}_i)\right],$$

where $\{\boldsymbol{\sigma}_i\}_{i=1}^N$ are sampled independently uniformly at random in $\{-1, 1\}$. Furthermore, if Δ is a random variable with values in $\mathcal{D}$, then the *Rademacher complexity of $\mathcal{F}$ with respect to Δ with N samples* is defined by

$$R_{\Delta,N}(\mathcal{F}) = E_{\boldsymbol{\delta}_1\sim\Delta,\dots,\boldsymbol{\delta}_N\sim\Delta}[R_D(\mathcal{F})]$$

wherein D stands for $\{\boldsymbol{\delta}_i\}_{i=1}^N$.

A well-known result in statistical learning states that if $D = \{\boldsymbol{\delta}_i\}_{i=1}^N$ is a set of N independent samples $\boldsymbol{\delta}_1 \sim \Delta, \dots, \boldsymbol{\delta}_N \sim \Delta$, then with confidence $1-\beta$ on the sampling of D, it holds that for every $f \in \mathcal{F}$, $\frac{1}{N}\sum_{i=1}^N f(\boldsymbol{\delta}_i)$ is "close" to $E_{\boldsymbol{\delta}\sim\Delta}[f(\boldsymbol{\delta})]$, where "close" is quantified with a quantity that depends on β, N and $R_{\Delta,N}(\mathcal{F})$. Formally, we have:

Proposition 1. (Mohri et al. 2018, Theorem 3.3). *It holds that*

$$P_{\boldsymbol{\delta}_1\sim\Delta,\dots,\boldsymbol{\delta}_N\sim\Delta}\left[\max_{f\in\mathcal{F}}\left\{E_{\boldsymbol{\delta}\sim\Delta}[f(\boldsymbol{\delta})] - \frac{1}{N}\sum_{i=1}^N f(\boldsymbol{\delta}_i)\right\} \le 2R_{\Delta,N}(\mathcal{F}) + \sqrt{\frac{\log(\frac{1}{\beta})}{2N}}\right] \ge 1-\beta. \tag{5}$$

8.2 Rademacher Complexity for Surrogate Constraint

We apply the result in Proposition 1 to the surrogate optimisation problem in Eq. (3). For that, we define $\mathcal{D}$ as the domain of Δ and $\mathcal{F} = \{\boldsymbol{\delta} \mapsto \mathbf{1}_{g(\boldsymbol{x},\boldsymbol{\delta})\le 0} \mid \boldsymbol{x} \in X\} \subseteq \mathbb{R}^{\mathcal{D}}$. It holds that for each $\boldsymbol{x} \in X$ and $D = \{\boldsymbol{\delta}_i\}_{i=1}^N \subseteq \mathcal{D}$, $E_{\boldsymbol{\delta}\sim\Delta}[\mathbf{1}_{g(\boldsymbol{x},\boldsymbol{\delta})\le 0}] = P[G_{\boldsymbol{x}} = 1]$ and $\sum_{i=1}^N \mathbf{1}_{g(\boldsymbol{x},\boldsymbol{\delta}_i)\le 0} = s_N(\boldsymbol{x}; D)$. Hence, we get the following:

*A more detailed explanation of the basis functions and their derivation can be found in the appendix of Jankowski et al. (2022).

†We use the notation $f'(s)$ for derivatives w.r.t. s and the notation $\dot{f}(s)$ for derivatives w.r.t. t.

Corollary 1. *With $\mathcal{F}$ defined as above, it holds that*

$$P_{\boldsymbol{\delta}_1\sim\Delta,\ldots,\boldsymbol{\delta}_N\sim\Delta}\left[\max_{\boldsymbol{x}\in X}\left\{P[G_{\boldsymbol{x}}=1]-\frac{1}{N}s_N(\boldsymbol{x};D)\right\}\right.$$
$$\left.\leq 2R_{\Delta,N}(\mathcal{F})+\sqrt{\frac{\log(\frac{1}{\beta})}{2N}}\right]\geq 1-\beta, \tag{6}$$

wherein D stands for $\{\boldsymbol{\delta}_i\}_{i=1}^N$.

Corollary 1 tells us that with confidence $1-\beta$ on the sampling of $D=\{\boldsymbol{\delta}_i\}_{i=1}^N$ with N i.i.d. samples from Δ, *any* solution $\boldsymbol{x}$ that is feasible for the surrogate optimisation problem Eq. (3) with

$$k_{\text{thresh}}\leq\left(\eta-2R_{\Delta,N}(\mathcal{F})-\sqrt{\frac{\log(\frac{1}{\beta})}{2N}}\right)N \tag{7}$$

is feasible for the original optimisation problem Eq. (1).

In view of Eq. (7), computing a suitable k_{thresh} for the surrogate optimisation problem in Eq. (3) can be approached by computing an upper bound on the Rademacher complexity of the associated set of functions $\mathcal{F}$. Despite the theoretical appeal of the this approach, computing an upper bound on the Rademacher complexity can be very challenging in general, and there is usually no closed-from expression for such bounds. Nevertheless, we present here a tight bound for a special case of collision-avoidance problem.

8.3 A Special Case of Collision Avoidance

We consider a robot motion planning problem, where a ball-shaped robot has to avoid m ball-shaped obstacles with high probability across time instants $t_1,\ldots,t_H$.

For the sake of simplicity, we first focus on the case with one obstacle ($m=1$) and one time step ($H=1$), before generalising. Thus, we consider the problem of finding the position $\boldsymbol{x}\in X$ ($X\subseteq\mathbb{R}^n$) of the center of the robot at time t_1 such that $P_{\boldsymbol{\delta}\sim\Delta}(\|\boldsymbol{x}-\boldsymbol{p}(\boldsymbol{\delta})\|\geq r)\geq 1-\eta$, where $r>0$ is the combined radius of the obstacle and the robot, and $\boldsymbol{p}(\boldsymbol{\delta})\in\mathbb{R}^n$ is the position of the center of the obstacle at time t_1 under scenario $\boldsymbol{\delta}$. In the formulation of Eq. (1), we have

$$g(\boldsymbol{x},\boldsymbol{\delta})=r-\|\boldsymbol{x}-\boldsymbol{p}(\boldsymbol{\delta})\|,$$

i.e., $g(\boldsymbol{x},\boldsymbol{\delta})\leq 0\Leftrightarrow\|\boldsymbol{x}-\boldsymbol{p}(\boldsymbol{\delta})\|\geq r$.

Let $\mathcal{F}$ be defined as in the previous subsection, *i.e.*, $\mathcal{F}=\{\mathbf{1}_{g(\boldsymbol{x},\boldsymbol{\delta})\leq 0}\mid\boldsymbol{x}\in X\}$, with X and g as above. In the subsequent section with the additional proofs (Appendix 8.4), we bound $R_{\Delta,N}(\mathcal{F})$ as follows:

$$R_{\Delta,N}(\mathcal{F})\leq\sqrt{\frac{d\log\left(\frac{eN}{d}\right)}{2N}}, \tag{8}$$

where $d=n+1$ and e is Euler's number. We obtain the following:

Proposition 2. *In the setting defined above with $m=H=1$, if we define $k_{\text{rad}}(\beta,N,\eta)$ as*

$$k_{\text{rad}}(\beta,N,\eta)=\eta N-\sqrt{2dN\log\left(\frac{eN}{d}\right)}-\sqrt{\frac{N\log(\frac{1}{\beta})}{2}},$$

then any feasible solution of the surrogate optimisation problem in Eq. (3) with $k_{\text{thresh}}=k_{\text{rad}}(\beta,N,\eta)$ is feasible for the original optimisation problem in Eq. (2) with confidence $1-\beta$ on the sampling of D.

Proof. Consequence of Eqs. (7) and (8).

We now discuss the case of $m\in\mathbb{N}_{\geq 1}$ obstacles and $H\in\mathbb{N}_{\geq 1}$ time steps. In this case,

$$g(\boldsymbol{x},\boldsymbol{\delta})=\max_{\substack{j=1,\ldots,m\\k=1,\ldots,H}}r-\|\boldsymbol{q}(\boldsymbol{x},t_k)-\boldsymbol{p}_j(\boldsymbol{\delta},t_k)\|,$$

i.e., $g(\boldsymbol{x},\boldsymbol{\delta})\leq 0\Leftrightarrow\forall\, j\,\forall\, k\,\|\boldsymbol{x}(t_k)-\boldsymbol{p}_j(\boldsymbol{\delta},t_k)\|\geq r$, where $\boldsymbol{x}(t_k)$ ($X\subseteq\mathbb{R}^{nH}$) is the position of the center of the robot at time t_k and $\boldsymbol{p}_j(\boldsymbol{\delta},t_k)\in\mathbb{R}^n$ is the position of the center of the j^{th} obstacle at time t_k under scenario $\boldsymbol{\delta}$. We show in Appendix 8.4 that $R_{\Delta,N}(\mathcal{F})$ can be bounded as follows:

$$R_{\Delta,N}(\mathcal{F})\leq mH\sqrt{\frac{d\log\left(\frac{eN}{d}\right)}{2N}}, \tag{9}$$

where $d=n+1$ and e is Euler's number. Similarly to the above, we get the following:

Proposition 3. *In the setting defined above with $m\in\mathbb{N}_{\geq 1}$ and $H\in\mathbb{N}_{\geq 1}$, if we define $k_{\text{rad}}(\beta,N,\eta)$ as*

$$k_{\text{rad}}(\beta,N,\eta)=$$
$$\eta N-mH\sqrt{2dN\log\left(\frac{eN}{d}\right)}-\sqrt{\frac{N\log(\frac{1}{\beta})}{2}},$$

then any feasible solution of the surrogate optimisation problem in Eq. (3) with $k_{\text{thresh}}=k_{\text{rad}}(\beta,N,\eta)$ is feasible for the original optimisation problem in Eq. (2) with confidence $1-\beta$ on the sampling of D.

Proof. Consequence of Eqs. (7) and (9).

Remark 1. Note that, unlike other approaches in the literature, Proposition 3 does not rely on *Boole's inequality* to bound the joint probability of collision avoidance. Indeed, the use of Boole's inequality would amount to set the collision avoidance probability for *each time step and each obstacle* to $\eta'=\frac{\eta}{mH}$, so that the probability of collision *with at least one obstacle at at least one time step* is bounded from above by $\eta=\sum_{j,k}\eta'$. Furthermore, we would need to set the confidence for *each time step and each obstacle* to $1-\beta'$ with $\beta'=\frac{\beta}{mH}$, in order to guarantee with confidence $1-\beta=1-\sum_{j,k}\beta'$ that the chance constraint holds for each of them simultaneously. This would result in a value of k_{thresh} as follows:

$$k'_{\text{rad}}(\beta,N,\eta)=$$
$$\frac{\eta N}{mH}-\sqrt{2dN\log\left(\frac{eN}{d}\right)}-\sqrt{\frac{N\log(\frac{mH}{\beta})}{2}}.$$

This can be rewritten as

$$k'_{\text{rad}}(\beta,N,\eta)=$$
$$\frac{\eta N-mH\sqrt{2dN\log\left(\frac{eN}{d}\right)}-mH\sqrt{\frac{N\log(\frac{mH}{\beta})}{2}}}{mH},$$

The above shows that for any values of β, N and η for which $k_{\text{rad}}(\beta, N, \eta) \geq 0$, it holds that

$$k'_{\text{rad}}(\beta, N, \eta) \leq \frac{1}{mH} k_{\text{rad}}(\beta, N, \eta).$$

Hence, using $k_{\text{thresh}} = k_{\text{rad}}(\beta, N, \eta)$ in Eq. (3) is less conservative (by a "factor" mH) than using $k_{\text{thresh}} = k'_{\text{rad}}(\beta, N, \eta)$.

8.4 Additional Proofs

We start with Eq. (8), reminded in the proposition below:

Proposition 4. *In the setting defined in Sec. 8.3 with $m = H = 1$, it holds that*

$$R_{\Delta,N}(\mathcal{F}) \leq \sqrt{\frac{d \log\left(\frac{eN}{d}\right)}{2N}}, \tag{10}$$

wherein $d = n + 1$ and e is Euler's number.

Proof. Consider the set of functions $\mathcal{H} = \{2f - 1 \mid f \in \mathcal{F}\}$, *i.e.*,

$$\mathcal{H} = \{2 \cdot \mathbf{1}_{\|\boldsymbol{x} - \boldsymbol{p}(\boldsymbol{\delta})\| \leq r} - 1 \mid \boldsymbol{x} \in X\}, \tag{11}$$

which is essentially the same as $\mathcal{F}$ except that the functions take values in $\{-1, 1\}$ instead of $\{0, 1\}$,[‡] and the quantity

$$\Pi_{\mathcal{H}}(N) = \max_{(\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_N) \in \mathcal{D}^N} \#\{(h(\boldsymbol{\delta}_1), \ldots, h(\boldsymbol{\delta}_N)) \mid h \in \mathcal{H}\}.$$

By (Mohri et al. 2018, Corollary 3.8), it holds that

$$R_{\Delta,N}(\mathcal{H}) \leq \sqrt{\frac{2 \log \Pi_{\mathcal{H}}(N)}{N}}.$$

We will bound $\Pi_{\mathcal{H}}(N)$ by $\left(\frac{eN}{d}\right)^d$ by using Eq. (11). For that, we consider the set of functions $\mathcal{H}' = \{h_{\boldsymbol{x},r} \mid (\boldsymbol{x}, r) \in \mathbb{R}^n \times \mathbb{R}_{\geq 0}\} \subseteq \mathbb{R}^{\mathcal{P}}$ where $\mathcal{P} = \mathbb{R}^n$ and $h_{\boldsymbol{x},r}(\boldsymbol{p}) = 2 \cdot \mathbf{1}_{\|\boldsymbol{x} - \boldsymbol{p}\| \leq r} - 1$, which contains all *ball classifiers* in $\mathbb{R}^n$ since $h_{\boldsymbol{x},r}(\boldsymbol{p}) = 1$ if $\boldsymbol{p}$ is in the ball of centre $\boldsymbol{x}$ and radius r, and -1 otherwise. It follows that the VC dimension of $\mathcal{H}'$, defined as the largest N for which $\Pi_{\mathcal{H}'} = 2^N$, is $d = n + 1$.[§] Hence, by (Mohri et al. 2018, Corollary 3.18), it follows that $\Pi_{\mathcal{H}'}(N) \leq \left(\frac{eN}{d}\right)^d$. It is also straightforward to show that $\Pi_{\mathcal{H}}(N) \leq \Pi_{\mathcal{H}'}(N)$ since for every $\boldsymbol{x} \in X$ and every $\boldsymbol{\delta} \in \mathcal{D}$, $2 \cdot \mathbf{1}_{\|\boldsymbol{x} - \boldsymbol{p}(\boldsymbol{\delta})\| \leq r} - 1 = h_{\boldsymbol{x},r}(\boldsymbol{p}(\boldsymbol{\delta}))$. Hence,

$$R_{\Delta,N}(\mathcal{H}) \leq \sqrt{\frac{2d \log(eN/d)}{N}}.$$

Finally, from (Shalev-Shwartz and Ben-David 2014, Lemma 26.9), we get that

$$R_{\Delta,N}(\mathcal{F}) \leq \frac{1}{2} R_{\Delta,N}(\mathcal{H}) \leq \sqrt{\frac{d \log(eN/d)}{2N}},$$

concluding the proof.

Next, we consider Eq. (9) and prove the following:

Proposition 5. *In the setting defined in Sec.8.3 with $m \in \mathbb{N}_{\geq 1}$ and $H \in \mathbb{N}_{\geq 1}$, it holds that*

$$R_{\Delta,N}(\mathcal{F}) \leq mH \sqrt{\frac{d \log\left(\frac{eN}{d}\right)}{2N}}, \tag{12}$$

wherein $d = n + 1$ and e is Euler's number.

Table 1. Offline Planning Experiments for $\beta = 0.05$

η	η_{rad}	η_{binom}	$\hat{\eta}_{\text{avg}}$	$\hat{\eta}_{1-\beta}$	$\bar{\beta}$
0.05	n/a	0.01	0.0218	0.0499	0.0497
	n/a	0.038	0.0393	0.0503	0.0539
0.1	n/a	0.04	0.0525	0.0936	0.0325
	n/a	0.084	0.0854	0.1012	0.0627
0.15	n/a	0.08	0.0934	0.1449	0.0375
	n/a	0.131	0.1322	0.1508	0.0572
0.2	n/a	0.13	0.1438	0.2060	0.0656
	n/a	0.178	0.1794	0.2010	0.0574
0.25	n/a	0.17	0.1842	0.2515	0.0535
	0.009	0.227	0.2288	0.2518	0.0638
0.3	n/a	0.22	0.2350	0.3068	0.0667
	0.059	0.275	0.2764	0.3005	0.0533
0.35	n/a	0.26	0.2755	0.3507	0.0517
	0.109	0.324	0.3251	0.3510	0.0578
0.4	n/a	0.31	0.3262	0.4053	0.0627
	0.159	0.374	0.3760	0.4029	0.0705
0.6	n/a	0.51	0.5282	0.6101	0.0754
	0.359	0.573	0.5748	0.6025	0.0683
0.8	0.158	0.72	0.7359	0.8053	0.0668
	0.559	0.778	0.7798	0.8024	0.0714

Proof. Note that by definition of g, it holds that

$$\mathbf{1}_{g(\boldsymbol{x},\boldsymbol{\delta})>0} = \max_{j,k} \mathbf{1}_{\|\boldsymbol{x}(t_k) - \boldsymbol{p}_j(\boldsymbol{\delta}, t_k)\| \leq r} - 1. \tag{13}$$

For each $j = 1, \ldots, m$ and $k = 1, \ldots, H$, let

$$\mathcal{F}_{j,k} = \{\mathbf{1}_{\|\boldsymbol{x}(t_k) - \boldsymbol{p}_j(\boldsymbol{\delta}, t_k)\| \leq r} - 1 \mid \boldsymbol{x} \in X\},$$

and let $\mathcal{F}' = \{\max_{j,k} f_{j,k} \mid \forall\, j\, \forall\, k\, f_{j,k} \in \mathcal{F}_{j,k}\}$. By Eq. (13), it holds that $\mathcal{F} \subseteq \mathcal{F}'$. By (Mohri et al. 2018, Ex. 3.8), it holds that $R_{\Delta,N}(\mathcal{F}') = \sum_{j,k} R_{\Delta,N}(\mathcal{F}_{j,k})$. Furthermore, by Proposition 4, we know that for each $j = 1, \ldots, m$ and $k = 1, \ldots, H$, $R_{\Delta,N}(\mathcal{F}_{j,k})$ is bounded from above by the right-hand side of Eq. (10). By summing over j and k, we get that $R_{\Delta,N}(\mathcal{F}')$ is bounded from above by the right-hand side term in Eq. (12). Since $\mathcal{F} \subseteq \mathcal{F}'$, we have that $R_{\Delta,N}(\mathcal{F}) \leq R_{\Delta,N}(\mathcal{F}')$, concluding the proof.

9 Additional Details on Naive vs. Confidence-Bounded Surrogate Constraint

We use Fig. 1 to illustrate the difference between the naïve formulation that only considers the maximum violation threshold η by setting $k_{\text{thresh}} = \eta N$ and the confidence-bounded formulation using k_β. For this purpose we analyse the binomial distribution with parameters N and $p = \eta = 0.1$ for different values of N, which correspond to the

[‡] This is done to stick to the classical theory of *binary classification* learning for which many results on the Rademacher complexity have been derived (Shalev-Shwartz and Ben-David 2014; Mohri et al. 2018).

[§] See for instance Sec. 15.5.2 in https://ti.inf.ethz.ch/ew/lehre/CG12/lecture/Chapter%2015.pdf (last consulted: July 30, 2024).

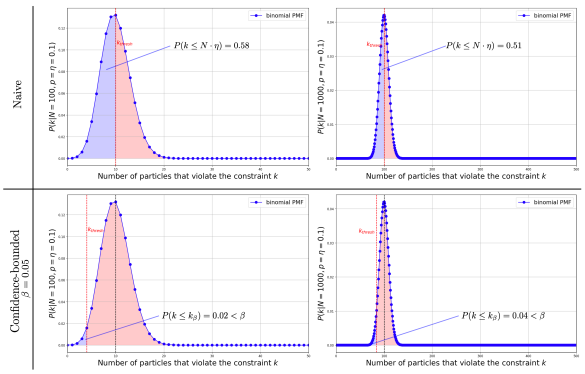


Figure 1. Analysis of the binomial distribution with $N = 100$ (left column) and $N = 1000$ (right column) Bernoulli experiments, which correspond to the number of samples used to approximate the chance constraint in the optimisation. The top row shows the values k_{thresh} takes for the naïve formulation of setting $k_{\text{thresh}} = \eta N$, where η is the user-defined maximum probability of violation. The bottom row shows the values k_{thresh} takes for the confidence-bounded formulation of the chance constraint, *i.e.*, $k_{\text{thresh}} = k_{\text{binom}}(\beta, N, \eta)$ for $\beta = 0.05$.

number of samples $\boldsymbol{\delta}_i \sim p_\Delta$ in the optimisation scheme. The plots in Fig. 1 shows the binomial distribution with $N = 100$ on the left and $N = 1000$ on the right. The blue shaded areas under the curve corresponds to the value of the CDF for k_{thresh}, *i.e.*, $P(K \leq k_{\text{thresh}} \mid N, \eta)$. The red-colored area under the curve corresponds to the the probability that $k > k_{\text{thresh}}$. The top row shows the naïve formulation, *i.e.*, setting $k_{\text{thresh}} = \eta N$. The bottom row shows the confidence-bounded formulation, *i.e.*, setting $k_{\text{thresh}} = k_{\text{binom}}(\beta, N, p)$ for $\beta = 0.05$.

10 Experiment Details

10.1 MPC Experiments: Environment Details

In this section, we provide additional details about the environments used for the MPC experiments in Sec. 6.1. We used three different environment configurations for which we generated the parameters randomly. Tab. 2 shows the specifications of the environments, *i.e.*, the number of obstacles N_{obs}, the initial obstacle positions $\boldsymbol{x}_0$, the initial obstacle velocities $\dot{\boldsymbol{x}}_0$, the obstacle radii and the variance of the obstacle accelerations $\ddot{\boldsymbol{x}}$, when sampling from a zero-mean Gaussian distribution in the random-walk model. The environment size was chosen to be consistent across all environments on a 10 by 10 grid.

10.2 Detailed Results on Offline-CC-VPSTO

The numerical results for the offline planning experiments are shown in Tab. 1.

10.3 Robot Experiment: Implementation of Stochastic Model

In this section, we provide additional details about the implementation of the stochastic model for the robot experiment in Sec. 6.2 describing the motion of the box obstacle on the conveyor belt. As our approach is Monte Carlo-based, in every MPC step we simulate the motion of the box obstacle for N_{sim} samples. The samples are initialised with the same position and velocity as the box obstacle at the beginning of the MPC step. The samples are then propagated through the conveyor belt dynamics for the duration of the MPC step. The conveyor belt dynamics are modelled as a probabilistic system, where the probability of changing direction increases over time. A sample at time step k is modeled by state vector $\mathbf{s} = [x_k, \dot{x}_k, p_k]$ where x_k is the position, $\dot{x}_k$ is the velocity, and p_k is the probability of changing direction at time step k. The dynamics of this system for each time step Δt can be described as follows:

1. *Update the Probability of Direction Change:*

$$p_{k+1} = p_k \cdot (1 - \alpha) \tag{14}$$

   where α is the rate at which the probability of a direction change increases over time.

2. *Determine the Direction Change:*

   - Sample a random number r from a uniform distribution between 0 and 1.
   - If $r < p_{k+1}$ or if the projected position $x_k + \dot{x}_k \Delta t$ is outside the boundaries of the conveyor belt, a direction change occurs.

3. *Update State based on Direction Change:*

$$\dot{x}_{k+1} = \begin{cases} -\dot{x} & \text{if direction change occurs} \\ \dot{x} & \text{otherwise} \end{cases} \tag{15}$$

$$p_{k+1} = \begin{cases} \alpha & \text{if direction change occurs} \\ p_{k+1} & \text{otherwise} \end{cases} \tag{16}$$

4. *Update Position:*

$$x_{k+1} = x + \dot{x}_{k+1} \Delta t \tag{17}$$

Therefore, the updated state vector after each time step is:

$$\mathbf{s}_{k+1} = [x_{k+1},\, \dot{x}_{k+1},\, p_{k+1}] \tag{18}$$

In summary, the above models the probabilistic dynamics of one box sample on the conveyor belt, where the direction of motion can change randomly influenced by the parameter α and the physical constraints of the system.

References

Jankowski J, Racca M and Calinon S (2022) From Key Positions to Optimal Basis Functions for Probabilistic Adaptive Control. *IEEE Robotics and Automation Letters* 7(2): 3242–3249.

Mohri M, Rostamizadeh A and Talwalkar A (2018) *Foundations of machine learning*. MIT press.

Shalev-Shwartz S and Ben-David S (2014) *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Zhang Z, Tomlinson J and Martin C (1997) Splines and linear control theory. *Acta Math. Appl* 49: 1–34.

Table 2. MPC Environment Specifications

Env.	0	1	2
N_{obs}	5	4	5
Robot radius	0.25	0.5	0.5
$\boldsymbol{x}_0$	$\begin{bmatrix} \begin{bmatrix} 2.0 \\ 4.0 \end{bmatrix}, \begin{bmatrix} 3.5 \\ 8.0 \end{bmatrix}, \begin{bmatrix} 7.5 \\ 2.5 \end{bmatrix}, \begin{bmatrix} 9.0 \\ 1.5 \end{bmatrix}, \begin{bmatrix} 4.5 \\ 8.0 \end{bmatrix} \end{bmatrix}$	$\begin{bmatrix} \begin{bmatrix} 7.9 \\ 5.7 \end{bmatrix}, \begin{bmatrix} 1.3 \\ 3.5 \end{bmatrix}, \begin{bmatrix} 4.9 \\ 9.4 \end{bmatrix}, \begin{bmatrix} 5.2 \\ 3.0 \end{bmatrix} \end{bmatrix}$	$\begin{bmatrix} \begin{bmatrix} 2.1 \\ 3.1 \end{bmatrix}, \begin{bmatrix} 6.8 \\ 5.0 \end{bmatrix}, \begin{bmatrix} 7.3 \\ 6.7 \end{bmatrix}, \begin{bmatrix} 4.2 \\ 4.2 \end{bmatrix}, \begin{bmatrix} 8.5 \\ 2.8 \end{bmatrix} \end{bmatrix}$
$\dot{\boldsymbol{x}}_0$	$\begin{bmatrix} \begin{bmatrix} 0.7 \\ 0.0 \end{bmatrix}, \begin{bmatrix} 0.25 \\ -0.5 \end{bmatrix}, \begin{bmatrix} -0.5 \\ 0.5 \end{bmatrix}, \begin{bmatrix} -0.1 \\ 0.1 \end{bmatrix}, \begin{bmatrix} 0.0 \\ -1.0 \end{bmatrix} \end{bmatrix}$	$\begin{bmatrix} \begin{bmatrix} 0.6 \\ 0.1 \end{bmatrix}, \begin{bmatrix} 0.0 \\ 0.2 \end{bmatrix}, \begin{bmatrix} -0.4 \\ 0.1 \end{bmatrix}, \begin{bmatrix} -0.2 \\ 0.0 \end{bmatrix} \end{bmatrix}$	$\begin{bmatrix} \begin{bmatrix} 0.5 \\ -0.2 \end{bmatrix}, \begin{bmatrix} 0.5 \\ 0.0 \end{bmatrix}, \begin{bmatrix} 0.0 \\ -0.2 \end{bmatrix}, \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix}, \begin{bmatrix} 0.2 \\ -0.3 \end{bmatrix} \end{bmatrix}$
Radii	$[0.5, 0.4, 0.3, 0.35, 0.55]$	$[-0.32, 0.51, 0.49, 0.34]$	$[0.54, 0.45, 0.55, 0.35, 0.34]$
$var(\dot{\boldsymbol{x}})$	$[0.5, 0.75, 0.65, 0.8, 0.6]$	$[0.54, 0.64, 0.51, 0.8]$	$[0.64, 0.66, 0.62, 0.57, 0.75]$

C

Appendices of Chapter 6

Touch-Based Object Localisation with Spatially-Aware Belief Entropy Estimation

Lara Brudermüller[1], Julius Jankowski[2], Marc Toussaint[3], and Nick Hawes[1]

***Abstract*— Robust robotic manipulation in the real world requires coping with incomplete or unreliable sensory input. While vision provides rich information, it often fails in the presence of occlusions, clutter, or poor lighting. In such cases, touch offers a robust alternative, enabling object localisation through contact alone. We present a touch-only global localisation method that operates in continuous state space with a particle belief. Sparse contact/no-contact signals are turned into informative likelihoods via a proximity-aware measurement model, and contact-aware resampling mitigates particle starvation. An information-gathering controller selects actions that maximise expected information gain using a non-parametric entropy estimator sensitive to both observation updates and dynamics. On real hardware, the system reliably localises and then grasps from broad, multi-modal initial beliefs with mode separations up to 0.4 m, far beyond the narrow uncertainty ranges assumed in related work. Information-aware localisation-actions sped up belief convergence and boost grasp success; and ablations in simulation confirm the benefits of the measurement and resampling components.**

I. Introduction

Humans can manipulate objects using only touch, even in the absence of vision. For robots to approach similar capabilities in unstructured or visually-challenging environments, such as those involving occlusions, clutter, or poor lighting, localising objects through touch alone represents a promising alternative. This has driven the development of algorithms that refine an object's pose estimate through deliberate physical interaction [1]–[3]. As an illustrative example, consider a robot having to retrieve a keyring from inside a bag using a multi-fingered hand. The object pose is initially unknown, contacts are intermittent and ambiguous, and the object may move such that exploratory interactions can either resolve ambiguity or further increase uncertainty in the object state. Such examples highlight the challenges of high-dimensional contact-rich estimation problems, where the robot often starts with little information. Estimating the posterior over object pose in these settings is computationally demanding: the complexity grows rapidly with both the number of degrees of freedom (DOFs) and the size of the initial uncertainty region [4]. As a result, most prior approaches restrict either the problem dimensionality or the scale of initial uncertainty. Yet, contact-rich manipulation is not only characterized by high-dimensional state and action spaces, but also the inherent highly non-linear contact dynamics and the multi-modality of the system state distributions [5].

[1]Oxford Robotics Institute, University of Oxford, UK; {larab, nickh}@robots.ox.ac.uk
[2]Amazon Robotics, Germany; jankowski@amazon.com
[3]TU Berlin, Germany; toussaint@tu-berlin.de

Fig. 1. Experimental setup of blind grasping. *Left to right:* initial particle belief with uniform weights (■); information-gathering trajectory rejecting particle hypotheses (■); converged belief after contact; successful grasp.

Discretising the problem space addresses the multi-modality while enabling standard filtering and planning over finite hypotheses [3], [6], but the curse of dimensionality limits the respective resolution and thus fails to capture the rich contact dynamics present in such manipulation tasks. Instead, these tasks require a framework that plans *in continuous state space*, anticipates *what* will be sensed and *how* actions reshape uncertainty through interaction, and operates from *uninformed, non-parametric* priors with broad support.

Towards this end, we address touch-based localisation with a system operating directly in continuous state spaces and actively shapes the belief distribution through contact. We propose a particle filter with a proximity-aware measurement model to turn sparse binary proprioceptive contact signals into informative likelihoods combined with contact-aware resampling. An information-gathering controller predicts how actions reshape the belief and selects trajectories that maximise expected information gain using a non-parametric entropy estimator that not only considers the probabilities of different object poses but also their spatial density.

Contributions: We address continuous-space object localisation through contact from uninformed, non-parametric beliefs, making the following contributions:

1) A *touch-only global localisation system* that plans and estimates directly in continuous state space with beliefs from uninformed, non-parametric priors, suitable for operation when vision is unreliable or absent.
2) A *proximity-aware measurement model* for contact that converts sparse binary signals into informative likelihoods, and a *contact-aware resampling* strategy that mitigates particle starvation under discontinuous observations.
3) A *sampling-based information-gathering controller* that selects candidate probing actions based on a non-parametric *differential entropy estimator* that captures both observation-driven changes (weights) and dynamics-driven changes (spatial density) in the belief.

On real hardware (cf. setup in Fig. 1) and in simulation, the

APPENDIX

A. Algorithmic Implementation Details

a) Contact Measurement Signal: We infer contact measurements from the robot's torque sensors. However, the Franka robot's torque sensors are too noisy to use directly, so we convert the signals into a binary contact indicator. Using an observer [1], we filter out torques due to gravity, friction, and actuation. We then compute the norm of the filtered torques from the first five joints, excluding the last two due to high noise, and apply a threshold to detect contact. This binary signal $z_t \in \{0, 1\}$ serves as an observation for the particle filter.

b) Low-level Control: We ensure moderate contact forces throughout the robot operation via an impedance controller on the low-level. While the control gains are higher during the localisation phase, we reduce the gains in the moment of grasping to allow for a more robust grasp. In parallel, we keep track of the contact forces acting on the robot's end effector by projecting the torques measured in the robot's joints onto the end effector frame. If the contact forces exceed a given threshold, the robot stops the current action early and transitions to the particle filter update phase.

c) Initial Belief: The initial belief about the object pose is represented as a set of particles, where each particle is a 6D pose of the object. The particles are sampled from a Gaussian mixture model with two components. For each component, we use a standard deviation of σ_{pos}=0.06 [m] for the box position and a standard deviation of σ_{ori}=0.5 [rad] for the yaw orientation. All weights are set to $1/N_p$, where N_p=100 is the number of particles.

d) Estimation of the Probability of Grasp Success: The probability of grasp success is estimated based on the maximum likelihood estimate of the object pose based on the updated belief after the localisation phase. We then compute a grasp primitive on the estimated object pose and simulate it on the full particle belief. The probability of grasp success is then computed as the ratio of the number of particles where the grasp was successful, i.e. the z-position of the object particles is above the ground, to the total number of particles, weighted by the respective particle weights.

e) Hyperparameters: In all experiments, the threshold on the probability of grasp success is set to 0.8. The number of candidates sampled in each iteration of the predictive sampling-based planner is set to N_{samples}=20. For our hardware experiments, this corresponded to the maximum number of threads that could be run in parallel. The support Ω for the entropy approximation was computed using $\rho = 5$ neighbours across all experiments. We have found that value to provide a good trade-off between local and global density, which has also been a good value in the quantitative experiments on the theoretical distributions provided in the supplementary material. The parameters for the proximity-based measurement model are set to $\alpha_{tp} = 0.5$, $\alpha_{fp} = 0.1$ and $\gamma = 1000$. Moreover, in the experiments, we set the maximum number of iterations, i.e. a maximum number of localising actions, to 15. After this number of iterations, the robot stops the localisation phase and transitions to the grasping phase, regardless of the probability of grasp success.

REFERENCES

[1] G. Garofalo, N. Mansfeld, J. Jankowski, and C. Ott, "Sliding mode momentum observers for estimation of external torques and joint acceleration," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6117–6123.