Stochastic Motion Planning and Control under Uncertainties

A Dissertation Presented to
The Academic Faculty

By

Hongzhe Yu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology

[December, 2025]

STOCHASTIC MOTION PLANNING AND CONTROL UNDER UNCERTAINTIES

Thesis committee:

Dr. Yongxin Chen
Professor
School of Aerospace Engineering, Georgia Institute of Technology

Dr. Samuel Coogan
Professor
School of Electrical and Computer Engineering, Georgia Institute of Technology

Dr. Panagiotis Tsiotras
Professor
School of Aerospace Engineering, Georgia Institute of Technology

Dr. Ye Zhao
Professor
School of Mechanical Engineering, Georgia Institute of Technology

Dr. Aaron Johnson
Professor
Department of Mechanical Engineering, Carnegie Mellon University

Date approved: December 5, 2025

TABLE OF CONTENTS

List of Tables v
List of Figures vi
Published Content and Contributions ix
Summary x
Chapter I: Introduction 1
Chapter II: Gaussian Variational Inference Motion Planning 5
2.1 Introduction 5
2.2 Trajectory Optimization Formulation 6
Continuous-Time Problem 6
Time Discretization and Support States 6
Sparse Precision Structure for Gaussian Processes 7
2.3 Planning as Probabilistic Inference 7
Gaussian Process Motion Planning (GPMP) Background 7
Variational Inference Formulation 8
Objective Function and Entropy Regularization 8
2.4 Natural Gradient Descent Algorithm 9
Parameterization and Gradients 9
Natural Gradient Update Rule 9
Line Search and Convergence 10
2.5 Sparse Factor Graph and Marginal Updates 10
Factor Graph Structure 10
Gradient Computation via Marginal Factorization 11
Closed-Form Updates for Prior Factors 13
Computational Complexity 14
2.6 Proximal Covariance Steering for Nonlinear Systems 14
Distributional Control Formulation 14
KL Divergence Reformulation and Duality 14
Proximal Gradient Algorithm in Distribution Space 15
Iterative Linear Approximation 16
Comparison: GVIMP vs. Proximal Covariance Steering 16
2.7 Experimental Validation 17
Point Robot Navigation 17
High-Dimensional Manipulator Planning 19
Computational Performance Analysis 19
Robustness to Uncertainties 20
Planar Quadrotor Example 21
2.8 Chapter Summary 22
Chapter III: Parallel Gaussian Variational Inference Motion Planning 31
3.1 KL-Proximal Variational Inference for Motion Planning 31

Proximal Point Algorithm with Splitting 32
Connection to Natural Gradient Descent 32
3.2 Distributed Computation on Sparse Factor Graphs 33
Gaussian Belief Propagation for Marginals 33
Parallel Collision Checking on GPU 34
3.3 Iterative P-GVIMP for Nonlinear Systems 34
Statistical Linear Regression 34
Algorithm Summary 35
3.4 Experimental Validation 35
Computational Efficiency 35
High-Dimensional Robot Arm Planning 39
Nonlinear Planar Quadrotor 40
3.5 Discussion and Connections 41
3.6 Chapter Summary 43
Chapter IV: Stochastic Control for Hybrid Systems 44
4.1 Introduction 44
4.2 Hybrid Dynamical Systems and Problem Formulation 45
Hybrid System Definition 45
Stochastic Flows and Transitions 45
Saltation Matrices and Uncertainty Propagation 46
Assumptions and Notations 47
4.3 Hybrid Covariance Steering 47
Problem Formulation 47
Path Measure Duality 48
Covariance Propagation Through Jumps 48
Convex Reformulation via Semidefinite Programming 49
Controller Recovery 49
4.4 Experiments 50
Experimental Validation: Bouncing Ball 50
Experimental Validation: SLIP Model 51
4.5 Hybrid Path Integral Control 52
Motivation and Overview 52
Nonlinear Hybrid Dynamics 53
Hybrid Stochastic Optimal Control Problem 53
Girsanov Theorem for Hybrid Processes 54
Cross-Entropy Formulation 54
Path Integral Optimal Controller 55
Importance Sampling and Variance Reduction 55
Experimental Validation: Bouncing Ball 56
Experimental Validation: SLIP Dynamics 57
Importance Sampling Efficiency Analysis 58
Comparison with Zero Control Proposal 59
4.6 Discussion and Connections 60
Unified Path-Distribution Perspective 60
Connections to Literature 61

Limitations and Future Directions 61
4.7 Chapter Summary 62
Chapter V: Conclusion and Future Work 63
5.1 Summary of Contributions 63
Gaussian Variational Inference Motion Planning 63
Parallel Gaussian Variational Inference Motion Planning 64
Stochastic Control for Hybrid Systems 64
5.2 Key Theoretical Insights 65
Duality Between Control and Inference 65
Sparse Structure and Computational Efficiency 65
Proximal Methods in Distribution Space 65
5.3 Limitations and Open Questions 65
5.4 Future Research Directions 66
GPU-Native Implementation 66
Diffusion-Based Variational Motion Planning 66
5.5 Concluding Remarks 67
Bibliography 69

LIST OF TABLES

Number *Page*

2.1 Time to compute prior costs in GVI-MP ($N = 50$ states; averaged over 50 runs). For 7-DOF, full-grid quadrature is computationally infeasible; sparse-grid is omitted as it would require 28^3 evaluations per iteration. 19

2.2 Optimization time for 7-DOF WAM experiments (averaged over 50 runs, $N = 50$). 20

2.3 Sampling time for drawing 1000 trajectories from the optimized distribution (7-DOF WAM, averaged over 50 runs). GPMP2 has no distribution to sample from. 20

2.4 Average minimum signed distance to disturbed obstacles over 50 trials. Positive values indicate collision-free execution; negative values indicate penetration. GVIMP demonstrates superior robustness. . . . 21

3.1 Collision checking time comparison (quadrature degree $k_q = 6$ for WAM, $k_q = 10$ for others). 36

3.2 Marginal covariance computation time (precision matrix dimension: 5000). 37

3.3 End-to-end optimization time comparison for complete motion planning problems ($N = 750$). 38

3.4 Average minimum distance from obstacles over 50 randomly perturbed environments. Negative values indicate collisions. 40

3.5 Comparison between GVIMP and P-GVIMP frameworks. 42

4.1 Expected cost improvement over H-iLQR baseline (100 trials). 57

4.2 Conditional Value at Risk (CVaR) cost improvement at different confidence levels. 57

4.3 Average weight variance and effective sample size before/after jump events. 58

LIST OF FIGURES

Number *Page*

2.1 GVIMP factor graph showing prior factors (connecting consecutive states) and collision factors (unary potentials at each state). The sparse connectivity enables efficient marginal computation. 11

2.2 Visual demonstration of the control-inference duality underlying both GVIMP and PCS-MP. Left: Stochastic optimal control formulation with explicit dynamics and costs. Right: Equivalent probabilistic inference formulation where planning becomes posterior inference under a cost-augmented prior. 17

2.3 Comparison of motion planning approaches on a 2-link arm. (a) GPMP2: deterministic MAP trajectory. (b) GVI-MP: maximum-entropy distributional solution with broader uncertainty. (c) PCS-MP: covariance-steered solution with prescribed terminal uncertainty. (d) Configuration space showing GPMP2 path (cyan) and 3σ uncertainty ellipses for GVI-MP (blue) and PCS-MP (red). 18

2.4 2D point-robot planning with $N = 50$ support states. GVI-MP yields broader distributions reflecting robustness to uncertainties, while PCS-MP achieves tighter covariance control. 25

2.5 WAM arm problem settings. Black spheres denote collision-checking points with safety radius r. 26

2.6 7-DOF WAM arm planning visualized in ROS MoveIt. Top row: Task 1. Bottom row: Task 2. All methods successfully navigate the bookshelf while maintaining collision-free trajectories. 26

2.7 2D point-robot resampling under obstacle perturbations. Original plan (blue) and resampled trajectories (red) from the optimized covariance. Resampling enables reactive adaptation without full replanning. 27

2.8 Franka hardware experiment settings. Scenario 1 features a single box obstacle. Scenario 2 includes additional non-convex constraints near the start configuration. 28

2.9 Hardware execution under disturbed obstacles (one representative trial per scenario). Top: Scenario 1. Bottom: Scenario 2. Baseline planners collide with perturbed obstacles; GVIMP remains safe through distributional planning and adaptive resampling. Additional results: https://www.youtube.com/watch?v=c4sFOlEki0Q . . . 29
2.10 Planar quadrotor planning via iterative linearization and GVIMP ($N = 3000$ states). Top: Four planning scenarios showing mean paths (solid lines) and 3σ position uncertainty (red ellipses). Bottom: Covariance evolution for velocity and angular state in Task 2, demonstrating uncertainty propagation through the dynamics. 30
3.1 Computation time for expected collision cost evaluation as a function of trajectory discretization N for the 7-DOF WAM arm. The parallel GPU implementation (blue) scales much more favorably than the serial CPU implementation (orange). 36
3.2 Computation time for marginal covariance extraction as a function of precision matrix dimension. Brute-force inversion (orange) scales cubically, while Gaussian Belief Propagation (blue) scales linearly. . . 37
3.3 P-GVIMP results for 2D point robot planning with four different start-goal configurations ($N = 50$ support states). Mean trajectories are shown as solid lines with 3σ covariance ellipses displayed at selected time steps. The algorithm finds collision-free paths while maintaining quantified uncertainty bounds. 38
3.4 WAM arm planning comparison for bookshelf task 1. GPMP2 produces a nominal trajectory (cyan), while P-GVIMP generates a trajectory distribution (gray mean with uncertainty ellipsoids). Both successfully avoid obstacles ($N = 750$ support states). 39
3.5 P-GVIMP results for PR2 right arm in two challenging environments ($N = 750$ support states). Gray trajectories show mean paths with uncertainty visualization. 40
3.6 Comparison of planners under perception noise. Green obstacles show nominal environment, yellow ones show perturbed positions. P-GVIMP and GPMP2 remain collision-free, while sampling-based planners collide with disturbed obstacles. 41
3.7 Iterative P-GVIMP result for nonlinear planar quadrotor. Red ellipses show 3σ position uncertainty along the trajectory. 42

4.1 Covariance steering for bouncing ball dynamics with elastic impacts. H-CS guarantees the terminal covariance constraint, while H-iLQR's covariance drifts from the target (black vs. colored trajectories). . . . 51
4.2 Deterministic nominal trajectory under H-iLQR controller (black) and stochastic sample trajectories under the H-CS controller (colored) for the SLIP model. The covariance tube narrows and expands according to the prescribed covariance profile. 52
4.3 Sample variance and effective sample portions for bouncing ball and SLIP. Both indicators change dramatically before and after jump events, revealing the impact of hybrid transitions on controller robustness. 59
4.4 Comparison of variance and effective samples for H-iLQR proposal (blue) versus zero control (orange). H-iLQR provides dramatically lower variance, enabling efficient importance sampling. 60

PUBLISHED CONTENT AND CONTRIBUTIONS

Chang, Zinuo et al. (2026). "Efficient iterative proximal variational inference motion planning". In: *Robotics and Autonomous Systems* 197, p. 105267. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2025.105267.

Hoshino, Kenta et al. (2025). "Path integral control of partially observed systems via fully observable control approximations". In: *Systems & Control Letters* 204, p. 106185.

Yu, Hongzhe and Yongxin Chen (2024). *Stochastic Motion Planning as Gaussian Variational Inference: Theory and Algorithms*. arXiv: 2308.14985 [cs.RO]. URL: https://arxiv.org/abs/2308.14985.

Yu, Hongzhe, Diana Frias Franco, et al. (2024a). "Optimal Covariance Steering of Linear Stochastic Systems with Hybrid Transitions". In: *arXiv preprint arXiv:2410.13222*.

– (2024b). "Path Integral Control for Hybrid Dynamical Systems". In: *arXiv preprint arXiv:2411.00659*.

Yu, Hongzhe and Yongxin Chen (2023). "A Gaussian variational inference approach to motion planning". In: *IEEE Robotics and Automation Letters* 8.5, pp. 2518–2525.

Yu, Hongzhe, Zhenyang Chen, and Yongxin Chen (2023). "Covariance steering for nonlinear control-affine systems". In: *arXiv preprint arXiv:2108.09530*.

Yu, Hongzhe, Joseph Moyalan, Umesh Vaidya, et al. (2022). "Data-driven optimal control of nonlinear dynamics under safety constraints". In: *IEEE Control Systems Letters* 6, pp. 2240–2245.

Yu, Hongzhe, Joseph Moyalan, Duvan Tellez-Castro, et al. (2021). "Convex optimal control synthesis under safety constraints". In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, pp. 4615–4621.

SUMMARY

Robotic systems operating in real-world environments must make decisions under pervasive uncertainties arising from imperfect models, sensor noise, actuator errors, and external disturbances. This dissertation develops a unified probabilistic framework for decision-making, motion planning, and control under uncertainty, grounded in stochastic optimal control and probabilistic inference.

At the core of the formulation is a stochastic optimal control problem for general nonlinear and hybrid dynamical systems with nonconvex cost functions. This formulation encompasses a wide range of robotic tasks while revealing two central challenges: (i) the computational intractability of solving stochastic control problems over high-dimensional trajectory distributions, and (ii) the need for algorithms that achieve both performance optimality and robustness to uncertainty.

To address these challenges, this dissertation introduces several novel methods: (1) *Gaussian Variational Inference Motion Planning (GVIMP)*, which frames motion planning as a variational inference problem in the space of trajectory distributions, providing a principled approach to approximate stochastic optimal control; (2) *Parallel Gaussian Variational Inference Motion Planning (P-GVIMP)*, an efficient proximal extension of GVIMP that exploits sparse factor-graph structures and Gaussian Belief Propagation (GBP) for GPU-parallelized gradient computation, enabling scalable planning for nonlinear stochastic systems; (3) an iterative *covariance steering* framework based on proximal gradient methods and iterative linearization, providing closed-loop solutions for nonlinear stochastic systems with prescribed uncertainty boundary conditions; and (4) two complementary algorithms for hybrid stochastic systems—*Hybrid Covariance Steering (H-CS)* for linear stochastic flows and *Hybrid Path Integral Control (H-PI)* for nonlinear flows—both derived from a

unified path-distribution control formulation.

Together, these contributions establish a cohesive theoretical and computational foundation that bridges stochastic optimal control, variational inference, and hybrid dynamical systems. The proposed methods are validated across diverse robotic benchmarks, demonstrating robust and efficient motion planning under uncertainty. Future directions include receding-horizon extensions for hybrid systems and learning-based multi-modal variational motion planning.

Chapter *1*

INTRODUCTION

Uncertainties are inherent in nearly every aspect of robotic systems and their operating environments (Thrun, 2002; Thrun, 2000). They arise from imperfect modeling of system dynamics and environmental interactions (Horak, 1988; Swevers, Verdonck, and De Schutter, 2007), actuator nonlinearities and parameter errors (Mesbah, 2018; Van Damme et al., 2011; Johannink et al., 2019), sensor noise and perceptual limitations (Mallick, Das, and Majumdar, 2014; Nguyen, Izadi, and Lovell, 2012; Durrant-Whyte, 1988), partial observations (Lovejoy, 1991; Krishnamurthy, 2016), and external disturbances (W.-H. Chen et al., 2000; Mohammadi et al., 2013). Understanding, modeling, and mitigating these uncertainties are fundamental to enabling robots to operate efficiently and safely in the real world.

A purely deterministic model of a robot's dynamics and environment fails to capture this intrinsic stochasticity. A more expressive and realistic representation is a *probabilistic world model* (Thrun, 2002), where the robot's state, actions, and observations are random variables governed by probability distributions. Within this probabilistic framework, this dissertation investigates the following fundamental questions:

1. How can probabilistic beliefs be modeled within the context of dynamical systems and optimal control?
2. How can we measure the quality or optimality of a robot's decision under probabilistic assumptions?
3. How do uncertainties affect system performance and stability?
4. How can we design optimal decision-making strategies that explicitly account for uncertainty?

Motion Planning under Uncertainty. Motion planning is a core component of robotic decision-making (LaValle, 2006; González et al., 2015). Given an environment, an initial configuration, and a goal configuration, the planner seeks a feasible trajectory connecting them. The trajectory optimization paradigm (Ratliff et al.,

2009; Schulman et al., 2014) formulates this task as an optimization problem over admissible trajectories subject to system dynamics and environmental constraints. Optimality criteria—such as travel time, control effort, or safety—define the cost function to be minimized.

When uncertainties are incorporated into the planning process (Kalakrishnan et al., 2011), the problem becomes one of *stochastic optimal control* (Bertsekas and Shreve, 1996) within the broader framework of *probabilistic robotics* (Åström, 2012; Thrun, 2002). The robot's stochastic dynamics can be modeled by stochastic differential equations, and the cost function becomes a statistical expectation over the distribution of trajectories. Motion planning in cluttered environments is inherently multimodal, as multiple homotopy classes of trajectories may exist. Selecting among these candidate plans requires balancing optimality and robustness. To this end, we introduce a novel entropy-based measure of the proposal distribution that quantifies the trade-off between performance and robustness under uncertainty.

Planning as Probabilistic Inference. For linear Gaussian systems, stochastic optimal control admits a dual formulation as a probabilistic inference problem over trajectory distributions (Kappen, Gómez, and Opper, 2012; Botvinick and Toussaint, 2012; Y. Chen, Tryphon T Georgiou, and Michele Pavon, 2016a; Y. Chen, Tryphon T Georgiou, and Michele Pavon, 2016b). This duality connects optimal control to probabilistic graphical inference. Building on this insight, we formulate motion planning as a variational inference problem, where the optimal trajectory distribution is obtained by minimizing the Kullback–Leibler (KL) divergence to a target posterior distribution defined by the stochastic control objective. Restricting the optimization to the Gaussian family leads to the *Gaussian Variational Inference Motion Planning (GVIMP)* framework developed in this dissertation (Yu and Y. Chen, 2023; Yu and Y. Chen, 2024). GVIMP efficiently computes approximate posterior distributions over trajectories that account for nonlinear dynamics and collision costs.

Building upon GVIMP, we further introduce an efficient proximal variant, termed *Parallel Gaussian Variational Inference Motion Planning (P-GVIMP) (Chang et al., 2026)*. P-GVIMP reformulates motion planning under uncertainty as a stochastic optimal control problem and solves a variational inference problem in the space of Gaussian path distributions using a proximal iterative scheme. For linear-Gaussian stochastic dynamics, a proximal algorithm is derived to update the Gaussian proposal toward the optimal posterior iteratively. The principal computational chal-

lenge—evaluating gradients over dense trajectory representations—is addressed by exploiting the sparsity of the underlying planning factor graph and employing Gaussian Belief Propagation (GBP) for efficient parallel computation on Graphics Processing Units (GPUs). Leveraging these properties, P-GVIMP achieves significant acceleration in large-scale planning problems while maintaining theoretical consistency with the GVIMP framework. Furthermore, the proximal iteration is generalized to nonlinear stochastic systems using Statistical Linear Regression (SLR), where P-GVIMP serves as an efficient subroutine for linearized time-varying systems. This combination enables practical, high-dimensional motion planning under uncertainty with near-real-time performance on modern hardware.

Covariance Steering for Stochastic Systems. In stochastic dynamical systems, the state covariance quantifies the system's uncertainty under a given controller. Steering this covariance from an initial value to a desired terminal one constitutes an *uncertainty control* problem (Chen., T. Georgiou, and M. Pavon, 2016; Y. Chen, Tryphon T. Georgiou, and Michele Pavon, 2016d; Y. Chen, Tryphon T Georgiou, and Michele Pavon, 2018; Okamoto, Goldshtein, and Tsiotras, 2018; Okamoto and Tsiotras, 2019; Ridderhof, Okamoto, and Tsiotras, 2019). One practical approach is to linearize the nonlinear system iteratively, solving a sequence of linear covariance steering subproblems in closed form. In this work, we develop an iterative method for nonlinear control-affine systems with prescribed initial and terminal covariance constraints. Each iteration yields a linear covariance steering problem with closed-form, closed-loop solutions. The outer-loop optimization adopts a proximal gradient framework, which enjoys a provably linear convergence rate. We demonstrate the efficacy of this approach on collision-avoidance tasks for robotic systems.

Hybrid Stochastic Control. Many robotic systems exhibit hybrid dynamics comprising continuous-time flows and discrete-time transitions, such as walking (Collins and Ruina, 2005; Laszlo, Panne, and Fiume, 1996; Todd, 2013; Kuindersma et al., 2016), running (Clark et al., 2001; Hutter et al., 2011; Westervelt et al., 2018), and manipulation (Johnson, Burden, and Koditschek, 2016; Billard and Kragic, 2019). Uncertainty impacts both the smooth and discrete components of these systems, often destabilizing their limit cycles (Manchester, Tobenkin, et al., 2011; Manchester, Mettin, et al., 2011). Robust uncertainty control is thus essential for hybrid robotic systems (Tassa, 2011; Drnach and Y. Zhao, 2021).

In the second part of this dissertation, we extend stochastic optimal control to hybrid

systems with guard and reset constraints. We consider both linear and nonlinear stochastic dynamics interconnected through deterministic hybrid transitions. These stochastic flows and transitions together define a *hybrid path distribution* under a given controller. We then recast the hybrid stochastic control problem as one of path-distribution control and develop two complementary methods:

- **Hybrid Covariance Steering (H-CS):** for systems with linear stochastic flows, this method steers the mean and covariance along a nominal trajectory with guaranteed performance.
- **Hybrid Path Integral Control (H-PI):** for systems with nonlinear stochastic flows, this method derives the optimal control via forward sampling of stochastic trajectories subject to hybrid transitions.

Together, these frameworks—GVIMP, P-GVIMP, H-CS, and H-PI—constitute a unified theory and computational toolkit for probabilistic motion planning and control under uncertainty, bridging inference, optimization, and hybrid dynamics within a coherent stochastic framework.

This dissertation is organized as follows: Chapter 2 presents the Gaussian Variational Inference Motion Planning (GVIMP) framework and Proximal Covariance Steering (PCS-MP) for motion planning under uncertainty. In Chapter 3, we introduce Parallel GVIMP (P-GVIMP), an efficient GPU-accelerated extension for large-scale planning problems. Chapter 4 extends the framework to hybrid dynamical systems, presenting methods for both linear and nonlinear stochastic flows. Finally, Chapter 5 concludes the dissertation and discusses future research directions.

Chapter *2*

GAUSSIAN VARIATIONAL INFERENCE MOTION PLANNING

2.1 Introduction

Motion planning under uncertainty requires reasoning about entire trajectory distributions rather than single nominal paths. This chapter introduces the Gaussian Variational Inference Motion Planning (GVIMP) framework, which formulates motion planning as a probabilistic inference problem over trajectory distributions. By framing the optimal planning problem as variational inference with Gaussian proposals, GVIMP provides a principled approach to computing trajectory distributions that balance optimality, safety, and robustness under uncertainty.

The key insight underlying GVIMP is the duality between stochastic optimal control and probabilistic inference. For linear-Gaussian systems, the optimal trajectory distribution solving a stochastic control problem can be characterized as a posterior distribution that combines prior dynamics with task-specific cost functions. Rather than seeking a single optimal trajectory, GVIMP approximates this posterior using variational inference within the family of Gaussian distributions, yielding both a mean trajectory and quantified uncertainty bounds.

This chapter presents the GVIMP framework in three main parts. First, we establish the problem formulation, connecting trajectory optimization to probabilistic inference through the planning-as-inference paradigm (Sections 2.2–2.3). Second, we derive a natural gradient descent algorithm that exploits the sparse factor graph structure underlying motion planning problems, enabling efficient computation through marginal factorization (Sections 2.4–2.5). We also present an alternative proximal gradient perspective in distribution space that connects GVIMP to optimal transport and covariance steering (Section 2.6). Third, we validate the approach through extensive experiments on point robots, manipulator arms, and aerial vehicles, demonstrating both the quality of the computed plans and their robustness to real-world uncertainties (Section 2.7).

The remainder of this chapter is organized as follows. Section 2.2 introduces the trajectory optimization formulation and its discretization. Section 2.3 develops the planning-as-inference duality and presents the GVIMP variational objective. Section 2.4 derives the natural gradient descent algorithm. Section 2.5 details

the sparse factor graph structure and marginal updates. Section 2.6 presents the proximal covariance steering alternative formulation. Section 2.7 presents experimental validation across multiple robotic systems. Section 2.8 summarizes the key contributions.

2.2 Trajectory Optimization Formulation

We begin by formulating motion planning as a continuous-time trajectory optimization problem, then introduce the time discretization that enables practical computation.

Continuous-Time Problem

Trajectory optimization leverages optimal control theory and formulates motion planning problems as an optimization in the time window $t \in [0, T]$ as follows:

$$\begin{aligned} \min_{X_t, u_t} \ & J_0(X_t, u_t) \\ \text{s.t.} \ & g_i^c(X_t, u_t) \leq 0, \ i = 1, \ldots, N_g \\ & f_i^c(X_t, u_t) = 0, \ i = 1, \ldots, N_f. \end{aligned} \tag{2.1}$$

Problem (2.1) is an optimization over the continuous-time variables (X_t, u_t), where $X_t : [0, T] \to \mathbb{R}^n$ is the state function of time of dimension n, and $u_t : [0, T] \to \mathbb{R}^m$ represents the control signal function of dimension m. J_0 is a cost function that often integrates a running cost function of (X_t, u_t) over $[0, T]$. $\{g_i^c\}_{i=1}^{N_g}$, $\{f_i^c\}_{i=1}^{N_f}$ are the functions defining the inequality and equality constraints that the motion planning problem is subject to, respectively.

Time Discretization and Support States

To enable numerical computation, we discretize the continuous time horizon. We define the time discretization

$$\mathbf{t} \triangleq [t_0, \ldots, t_N], \ t_0 = 0, \ t_N = T \tag{2.2}$$

that generates a vector of discretized states and control inputs of length $N + 1$. We denote the *support states* and *support controls* as the discretized variables

$$\mathbf{x} \triangleq [X_0, \ldots, X_N]^T, \ \mathbf{U} \triangleq [u_0, \ldots, u_N]^T, \tag{2.3}$$

where $X_i = X_{t_i}, \ u_i = u_{t_i}, i = 1, \ldots, N$. When $\mathbf{x}$ is a random variable, the covariance matrix of $\mathbf{x}$ is denoted as $\mathbf{K}$. The discretized variables have the dimensions $\mathbf{x} \in \mathbb{R}^{(N+1)\times n}$, $\mathbf{U} \in \mathbb{R}^{(N+1)\times m}$, and $\mathbf{K} \in \mathbb{R}^{(N+1)n\times(N+1)n}$.

Sparse Precision Structure for Gaussian Processes

For linear Gauss-Markov processes, the joint precision matrix $\mathbf{K}^{-1}$ admits a sparse factorization (Tim D Barfoot, Tong, and Särkkä, 2014) that will be crucial for computational efficiency. Specifically, we have

$$\mathbf{K}^{-1} \triangleq G^T Q^{-1} G, \tag{2.4}$$

where

$$G = \begin{bmatrix} I & & & & \\ -\Phi(t_1, t_0) & I & & & \\ & & \cdots & & \\ & & & -\Phi(t_N, t_{N-1}) & I \\ & & & 0 & I \end{bmatrix}, \tag{2.5}$$

and

$$Q^{-1} = \text{diag}(K_0^{-1}, Q_{0,1}^{-1}, \ldots, Q_{N-1,N}^{-1}, K_N^{-1}). \tag{2.6}$$

Here $\Phi(t_{i+1}, t_i)$ denotes the state transition matrix, $Q_{i,i+1}$ is the process noise covariance (Grammian) on the interval (t_i, t_{i+1}), and K_0, K_N are penalty terms encoding boundary conditions at start and goal. This sparse tridiagonal block structure will enable efficient gradient computation in Section 2.5.

2.3 Planning as Probabilistic Inference

Motion planning admits a dual formulation as probabilistic inference (Toussaint, 2009; Mukadam, Yan, and Boots, 2016; Mukadam, Dong, et al., 2018). This section develops this duality and introduces the GVIMP variational objective.

Gaussian Process Motion Planning (GPMP) Background

Gaussian Process Motion Planning (GPMP) (Mukadam, Yan, and Boots, 2016) formulated the motion planning problem (2.1) as inference over a posterior probability:

$$p(\mathbf{x}|Z) \propto p(\mathbf{x}) p(Z|\mathbf{x}), \tag{2.7}$$

where $\mathbf{x}$ represents the discretized trajectory (support states), $p(\mathbf{x})$ encodes the prior dynamics as a Gaussian process, and $p(Z|\mathbf{x})$ represents the likelihood of satisfying task constraints (e.g., collision avoidance) given environment Z.

In the GPMP framework, the prior $p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$ captures the smooth flow induced by the system dynamics, where $\boldsymbol{\mu}$ is the mean trajectory under zero control and $\mathbf{K}$ is the covariance with the sparse inverse structure described in (2.4)–(2.6). The

likelihood $p(Z|\mathbf{x})$ typically takes the form $p(Z|\mathbf{x}) \propto e^{-c(\mathbf{x})}$, where $c(\mathbf{x})$ aggregates collision costs and other task-specific penalties.

GPMP solves for the maximum a posteriori (MAP) trajectory by minimizing $-\log p(\mathbf{x}|Z)$, which yields a deterministic solution. While effective, this approach does not quantify uncertainty in the resulting plan.

Variational Inference Formulation

Our formulation extends GPMP by seeking not a single trajectory but an entire distribution over trajectories. Rather than computing only the MAP estimate, GVIMP approximates the full posterior distribution (2.7) using variational inference. This distributional solution enables uncertainty quantification and robust execution strategies.

Let $\mathcal{Q} \triangleq \{q_\theta : q_\theta \sim \mathcal{N}(\mu_\theta, \Sigma_\theta)\}$ denote the family of Gaussian distributions parameterized by mean μ_θ and covariance Σ_θ. GVIMP seeks the optimal Gaussian approximation by solving:

$$q^\star = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}\left(q(\mathbf{x}) \parallel p(\mathbf{x}|Z)\right) \tag{2.8}$$

where $\mathrm{KL}(\cdot \parallel \cdot)$ denotes the Kullback–Leibler divergence (Van Erven and Harremos, 2014), a distributional distance measure between the proposal Gaussian q and the target posterior $p(\mathbf{x}|Z)$.

Equation (2.8) formulates probabilistic motion planning as an optimization within the Gaussian distribution space, where the objective function measures how well the proposal distribution approximates the true posterior. By restricting attention to Gaussian distributions, we obtain a tractable optimization problem while maintaining sufficient expressiveness to capture uncertainty in the planned trajectories.

Objective Function and Entropy Regularization

Define the negative log probability for the posterior as $\psi(\mathbf{x}) \triangleq -\log p(\mathbf{x}|Z)$. The objective in (2.8) can be rewritten as:

$$\mathcal{J}(q) = \mathrm{KL}\left(q(\mathbf{x}) \parallel p(\mathbf{x}|Z)\right) = \mathbb{E}_q[\psi(\mathbf{x})] - \mathcal{H}(q), \tag{2.9}$$

where $\mathcal{H}(q) = -\int q(\mathbf{x}) \log q(\mathbf{x}) d\mathbf{x}$ denotes the differential entropy of the distribution q. This decomposition reveals an important trade-off: minimizing $\mathbb{E}_q[\psi(\mathbf{x})]$ encourages low-cost trajectories, while maximizing $\mathcal{H}(q)$ promotes distributional diversity and robustness.

The entropy term $\mathcal{H}(q)$ plays a crucial role in robust planning. Higher entropy indicates greater uncertainty, which can be beneficial when facing model inaccuracies or environmental disturbances. By balancing cost and entropy, GVIMP naturally trades off nominal optimality against robustness. This connection to maximum-entropy reinforcement learning (Haarnoja et al., 2018; R. Zhao, Sun, and Tresp, 2019) provides a principled approach to risk-sensitive planning.

2.4 Natural Gradient Descent Algorithm

This section derives a natural gradient descent algorithm for solving the variational problem (2.8). Natural gradients (Amari, 1998) account for the Riemannian geometry of the parameter space, leading to faster and more stable convergence than standard Euclidean gradients.

Parameterization and Gradients

The sparsity of the Gaussian prior (via the precision matrix structure (2.4)) motivates parameterizing the proposal distribution using its mean μ_θ and precision Σ_θ^{-1} (inverse covariance). Taking derivatives of the objective (2.9) with respect to these parameters yields:

$$\frac{\partial \mathcal{J}(q)}{\partial \mu_\theta} = \Sigma_\theta^{-1}\mathbb{E}\left[(\mathbf{x}-\mu_\theta)\psi\right] \tag{2.10a}$$

$$\frac{\partial^2 \mathcal{J}(q)}{\partial \mu_\theta \partial \mu_\theta^T} = \Sigma_\theta^{-1}\mathbb{E}\left[(\mathbf{x}-\mu_\theta)(\mathbf{x}-\mu_\theta)^T\psi\right]\Sigma_\theta^{-1} - \Sigma_\theta^{-1}\mathbb{E}\left[\psi\right]. \tag{2.10b}$$

These expressions involve expectations under the current proposal distribution q_θ, which can be evaluated using the sparse factor graph structure developed in Section 2.5.

Natural Gradient Update Rule

Natural gradient descent accounts for the geometry of the distribution space by pre-conditioning the gradient with the Fisher information matrix. For Gaussian distributions, the natural gradient update takes the form:

$$\begin{bmatrix} \delta\mu_\theta \\ \mathrm{vec}(\delta\Sigma_\theta^{-1}) \end{bmatrix} = -\begin{bmatrix} \Sigma_\theta & 0 \\ 0 & 2(\Sigma_\theta^{-1}\otimes\Sigma_\theta^{-1}) \end{bmatrix}\begin{bmatrix} \frac{\partial \mathcal{J}(q)}{\partial \mu_\theta} \\ \mathrm{vec}(\frac{\partial \mathcal{J}(q)}{\partial \Sigma_\theta^{-1}}) \end{bmatrix},$$

which in matrix form simplifies to:

$$\Sigma_\theta^{-1}\delta\mu_\theta = -\frac{\partial \mathcal{J}(q)}{\partial \mu_\theta}, \quad \delta\Sigma_\theta^{-1} = -2\Sigma_\theta^{-1}\frac{\partial \mathcal{J}(q)}{\partial \Sigma_\theta^{-1}}\Sigma_\theta^{-1}. \tag{2.11}$$

Comparing (2.10) and (2.11), we observe that the precision update can be expressed directly in terms of the second-order gradient:

$$\delta\Sigma_\theta^{-1} = \frac{\partial^2 \mathcal{J}(q)}{\partial \mu_\theta \partial \mu_\theta^T} - \Sigma_\theta^{-1}. \tag{2.12}$$

Equations (2.11) and (2.12) show that computing the updates $\delta\mu_\theta, \delta\Sigma_\theta^{-1}$ requires only evaluating the expectations (2.10a) and (2.10b), which can be done efficiently using the marginal factorization described in Section 2.5.

Line Search and Convergence

The updates use a backtracking line search with step size $\eta < 1$:

$$\mu_\theta \leftarrow \mu_\theta + \eta^R \, \delta\mu_\theta, \quad \Sigma_\theta^{-1} \leftarrow \Sigma_\theta^{-1} + \eta^R \, \delta\Sigma_\theta^{-1}, \tag{2.13}$$

where $R = 1, 2, \ldots$ is increased to shrink the step size until the objective decreases. This ensures monotonic improvement and stable convergence.

The natural gradient algorithm exhibits superior convergence properties compared to standard gradient descent, particularly for ill-conditioned problems where the eigenvalues of the Hessian span multiple orders of magnitude. By accounting for the intrinsic geometry of the distribution space, natural gradients automatically adapt the step size along different parameter directions.

2.5 Sparse Factor Graph and Marginal Updates

The key to computational efficiency in GVIMP lies in exploiting the sparse structure of the underlying factor graph. This section shows how the gradient computations (2.10) can be decomposed across marginal distributions, enabling both theoretical analysis and efficient implementation.

Factor Graph Structure

The motion planning factor graph, illustrated in Figure 2.1, contains two types of factors:

- **Prior factors** $f_{i,i+1}(x_i, x_{i+1})$ encode the kinodynamic constraints, promoting smooth trajectories that respect the system dynamics.
- **Collision factors** $f_i(x_i)$ encourage the trajectory to remain within obstacle-free regions.

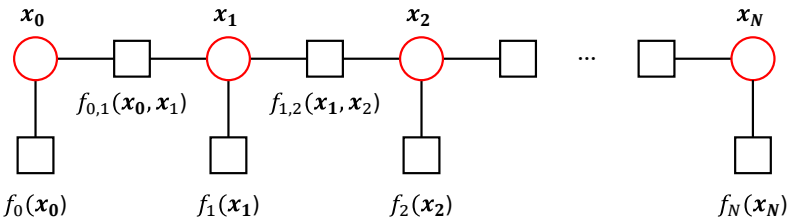


Figure 2.1. GVIMP factor graph showing prior factors (connecting consecutive states) and collision factors (unary potentials at each state). The sparse connectivity enables efficient marginal computation.

This factored structure reflects the locality of the planning problem: dynamics couple only adjacent time steps, and collision costs depend only on local configuration. By leveraging this sparsity, we can distribute the computation of expectations across the factor graph.

Gradient Computation via Marginal Factorization

Stein's Lemma. Let $X \sim \mathcal{N}(0, 1)$ and let $f : \mathbb{R} \to \mathbb{R}$ be a differentiable function such that $\mathbb{E}[\,|f'(X)|\,] < \infty$. Then

$$\mathbb{E}[\,Xf(X)\,] = \mathbb{E}[\,f'(X)\,]. \tag{2.14}$$

More generally, if $X \sim \mathcal{N}(\mu, \sigma^2)$, then

$$\mathbb{E}[\,(X-\mu)f(X)\,] = \sigma^2\,\mathbb{E}[\,f'(X)\,]. \tag{2.15}$$

Vector Stein's Lemma. Let $X \sim \mathcal{N}(0, I_d)$ and let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable with $\mathbb{E}[\|\nabla f(X)\|] < \infty$. Then

$$\mathbb{E}[\,Xf(X)\,] = \mathbb{E}[\,\nabla f(X)\,]. \tag{2.16}$$

More generally, for $X \sim \mathcal{N}(\mu, \Sigma)$,

$$\mathbb{E}[\,(X-\mu)f(X)\,] = \Sigma\,\mathbb{E}[\,\nabla f(X)\,]. \tag{2.17}$$

Componentwise, this is

$$\mathbb{E}\big[(X_i-\mu_i)f(X)\big] = \sum_{j=1}^{d} \Sigma_{ij}\,\mathbb{E}\left[\frac{\partial f}{\partial x_j}(X)\right]. \tag{2.18}$$

The gradient calculation (2.10) breaks down into computing expectations of $\psi(\mathbf{x})$ and its derivatives:

$$\mathbb{E}\left[\psi\right],\ \mathbb{E}\left[(\mathbf{x}-\mu_\theta)\,\psi\right],\ \mathbb{E}\left[(\mathbf{x}-\mu_\theta)\,(\mathbf{x}-\mu_\theta)^T\,\psi\right].$$

By Stein's Lemma, these expectations can be expressed in terms of gradients of ψ:

$$\nabla_{\mu_\theta}\psi(\mathbf{x},\theta) = \mathbb{E}\left[\nabla_{\mathbf{x}}\psi(\mathbf{x})\right], \tag{2.19a}$$

$$\nabla_{\Sigma_\theta}\psi(\mathbf{x},\theta) = \frac{1}{2}\mathbb{E}\left[\nabla^2_{\mathbf{xx}}\psi(\mathbf{x})\right]. \tag{2.19b}$$

Since the collision cost factorizes as $\psi(\mathbf{x}) = \sum_{i=0}^{N}\psi_i(\mathbf{x}_i)$, where $\mathbf{x}_i = M_i\mathbf{x}$ extracts the marginal state at time i via the selection matrix M_i, the gradients decompose as:

$$\psi(\mathbf{x}) = \sum_{i=0}^{N}\psi_i(\mathbf{x}_i) = \sum_{i=0}^{N}\left\|\mathbf{h}\left(\mathbf{x}_i\right)\right\|^2_{\Sigma_{\text{obs}_i}}, \tag{2.20}$$

where $\mathbf{h}(\cdot)$ measures signed distance to obstacles, and Σ_{obs_i} weights the collision penalty.

Applying the factorization (2.20) to (2.19a), we obtain:

$$\mathbb{E}_q\left[\nabla_{\mathbf{x}}\psi(\mathbf{x})\right] = \sum_{i=0}^{N} M_i^{\mathrm{T}}\mathbb{E}_{q_i}\left[\nabla_{\mathbf{x}_i}\psi_i\left(\mathbf{x}_i\right)\right], \tag{2.21}$$

where q_i denotes the marginal distribution of $\mathbf{x}_i$ under the joint proposal q. Similarly, for the second-order term:

$$\mathbb{E}_q\left[\nabla^2_{\mathbf{xx}}\psi(\mathbf{x})\right] = \sum_{i=0}^{N} M_i^{\mathrm{T}}\mathbb{E}_{q_i}\left[\nabla^2_{\mathbf{x}_i\mathbf{x}_i}\psi_i\left(\mathbf{x}_i\right)\right]M_i. \tag{2.22}$$

Applying Stein's Lemma again yields the final factorized gradient expressions:

$$\nabla_{\mu_\theta}\psi(\mathbf{x},\theta) = \sum_{i=0}^{N} M_i^{\mathrm{T}}\nabla_{\mu^i_\theta}\psi_i(\mathbf{x}_i,\theta_i), \tag{2.23a}$$

$$\nabla_{\Sigma_\theta}\psi(\mathbf{x},\theta) = \sum_{i=0}^{N} M_i^{\mathrm{T}}\nabla_{\Sigma^i_\theta}\psi_i(\mathbf{x}_i,\theta_i)M_i, \tag{2.23b}$$

where $\theta_i = (\mu^i_\theta, \Sigma^i_\theta)$ denotes the parameters of the marginal distribution q_i.

This factorization is crucial for computational efficiency: instead of computing expectations over the $(N+1)n$-dimensional joint distribution, we need only evaluate expectations over n-dimensional marginals. Furthermore, these marginal computations are independent and can be parallelized, as exploited in Chapter 3.

Closed-Form Updates for Prior Factors

The prior factor ψ_{Prior} contains the sparse precision matrix $\mathbf{K}^{-1}$ from (2.4). Denoting the discretized states as $X_j = X_{t_j}, \quad j = 0, \dots, N$, and using the state transition matrices $\Phi_{j+1,j}$, we have:

$$\begin{aligned}\psi_{\text{Prior}}(\mathbf{X}) &\propto \mathbb{E}\{(\mathbf{X}-\boldsymbol{\mu})^T G^T Q^{-1} G(\mathbf{X}-\boldsymbol{\mu})\} \\ &= \mathbb{E}\{\|[X_0^T, (X_1 - \Phi_{1,0}X_0)^T, \dots, (X_N - \Phi_{N,N-1}X_{N-1})^T, X_N^T]^T\|^2_{Q^{-1}}\}.\end{aligned}$$

This expression decomposes into marginal costs:

$$\psi_{\text{Prior}} = \mathcal{J}_0(q_0) + \mathcal{J}_{0,1}(q_{0,1}) + \cdots + \mathcal{J}_{N-1,N}(q_{N-1,N}) + \mathcal{J}_N(q_N),$$

where the factorized cost functionals are defined as:

$$\mathcal{J}_0 \triangleq \mathbb{E}_{q_0}\left[\|X_0 - \mu_0\|^2_{K_0^{-1}}\right], \tag{2.24a}$$

$$\mathcal{J}_{i,i+1} \triangleq \mathbb{E}_{q_{i,i+1}}\left[\|X_{i+1} - \Phi_{i+1,i}X_i\|^2_{Q^{-1}_{i,i+1}}\right], \tag{2.24b}$$

$$\mathcal{J}_N \triangleq \mathbb{E}_{q_N}\left[\|X_N - \mu_N\|^2_{K_N^{-1}}\right]. \tag{2.24c}$$

Here q_0 and q_N are the marginal Gaussian distributions of the initial and terminal states, and $q_{i,i+1}$ denotes the joint marginal of consecutive states at times t_i and t_{i+1}.

The factors $\mathcal{J}_{i,i+1}(q_{i,i+1})$ promote consistency with the uncontrolled dynamics, while $\mathcal{J}_0$ and $\mathcal{J}_N$ enforce the boundary conditions. Because the underlying dynamics are linear, all these factors have quadratic forms inside the expectations. This special structure enables closed-form gradient computation.

Lemma 1 (Closed-Form Prior Gradients)**.** *For a Gaussian random variable* $\mathbf{X}$ *and given matrices* Λ, Ψ *with appropriate dimensions, define* $y \triangleq \mathbf{x} - \mu_\theta$*. The derivatives of* $\mathcal{J}_{\Lambda,\Psi} \triangleq \mathbb{E}[\|\Lambda\mathbf{x} - \Psi\boldsymbol{\mu}\|^2_{\mathbf{K}^{-1}}]$ *with respect to the mean* μ_θ *are:*

$$\frac{\partial \mathcal{J}_{\Lambda,\Psi}}{\partial \mu_\theta} = 2\Lambda^T\mathbf{K}^{-1}(\Lambda\mu_\theta - \Psi\boldsymbol{\mu}), \tag{2.25a}$$

$$\frac{\partial^2 \mathcal{J}_{\Lambda,\Psi}}{\partial \mu_\theta \partial \mu_\theta^T} = \Sigma_\theta^{-1}\mathbb{E}\left[yy^T y^T \Lambda^T\mathbf{K}^{-1}\Lambda y\right]\Sigma_\theta^{-1} - \Sigma_\theta^{-1}\text{tr}\left[\Lambda^T\mathbf{K}^{-1}\Lambda\Sigma_\theta\right]. \tag{2.25b}$$

This lemma enables exact, efficient evaluation of prior factor contributions without requiring numerical quadrature. Combined with sparse quadrature methods (Heiss and Winschel, 2008) for the collision factors (where ψ_i is nonlinear), we obtain a hybrid approach that balances accuracy and efficiency.

Computational Complexity

The overall complexity of the GVIMP algorithm is $O(Nn^{k_q})$, where N is the number of discretization points, n is the state dimension, and k_q is the polynomial degree of the Gauss-Hermite quadrature used for collision cost expectations (Yu and Y. Chen, 2024). The linear dependence on N stems from the marginal factorization, while the polynomial dependence on n arises from the quadrature rule. For typical robotic systems with $n \leq 14$ (e.g., 7-DOF manipulators), using $k_q = 6$ provides sufficient accuracy.

2.6 Proximal Covariance Steering for Nonlinear Systems

While the natural gradient algorithm derived in Section 2.4 provides an effective approach for Gaussian variational inference, an alternative perspective emerges by formulating the problem directly in the space of path distributions. This section introduces a proximal gradient method that connects GVIMP to optimal transport theory and covariance steering, providing complementary insights into the structure of stochastic motion planning problems.

Distributional Control Formulation

Consider the stochastic optimal control problem with nonlinear dynamics and covariance boundary conditions:

$$\min_{u} \quad \mathbb{E}\left\{\int_0^T [\frac{1}{2}\|u_t\|^2 + V(X_t)]dt\right\} \tag{2.26a}$$

$$dX_t = f(t, X_t)dt + B(t)(u_t dt + \sqrt{\epsilon}dW_t) \tag{2.26b}$$

$$X_0 \sim \rho_0, \quad X_T \sim \rho_T, \tag{2.26c}$$

where ρ_0 (ρ_T) is a probability distribution with mean m_0 (m_T) and covariance Σ_0 (Σ_T). The cost function decouples into control effort $\frac{1}{2}\|u_t\|^2$ and state-dependent cost $V(X_t)$. Problem (2.26) has been studied in distribution control theory (Y. Chen, Tryphon T. Georgiou, and Michele Pavon, 2016c; Caluya and Halder, 2020), which links optimal control to optimal transport.

KL Divergence Reformulation and Duality

The connection between stochastic optimal control and Gaussian variational inference can be visualized through the duality shown in Figure 2.2. On the left, we have the explicit stochastic control formulation with dynamics and costs. On the right, the same problem is cast as probabilistic inference where planning becomes posterior inference under a cost-augmented prior.

Problem (2.26) can be reformulated as an optimization over path distributions:

$$\min_{\mathcal{P}^u} \quad \int d\mathcal{P}^u \left[\log \frac{d\mathcal{P}^u}{d\mathcal{P}^0} + \frac{1}{\epsilon} V\right] \tag{2.27a}$$

$$(X_0)_\sharp \mathcal{P}^u = \rho_0, \quad (X_T)_\sharp \mathcal{P}^u = \rho_T, \tag{2.27b}$$

where $\mathcal{P}^u$ denotes the path measure induced by the controlled process, $\mathcal{P}^0$ is the measure of the uncontrolled (prior) process, and $(X_t)_\sharp \mathcal{P}$ denotes the pushforward (marginal) distribution at time t.

Let

$$F(\mathcal{P}^u) = \int [\frac{1}{\epsilon} V - \log d\mathcal{P}^0] d\mathcal{P}^u$$

and

$$G(\mathcal{P}^u) = \int d\mathcal{P}^u \log d\mathcal{P}^u.$$

Then (2.27) becomes a composite optimization:

$$\min_{\mathcal{P}^u \in \Pi(\rho_0, \rho_T)} F(\mathcal{P}^u) + G(\mathcal{P}^u), \tag{2.28}$$

where $\Pi(\rho_0, \rho_T)$ denotes the set of path measures with prescribed initial and terminal marginals.

Proximal Gradient Algorithm in Distribution Space

The proposed algorithm employs a proximal gradient descent paradigm in the distribution space. Let $D(\cdot, \cdot)$ be a Bregman divergence; the generalized non-Euclidean proximal gradient iteration reads:

$$\mathcal{P}^{k+1} = \arg\min_{\mathcal{P} \in \mathcal{Y}} G(\mathcal{P}) + \frac{1}{\eta} D(\mathcal{P}, \mathcal{P}^k) + \langle \nabla F(\mathcal{P}^k), \mathcal{P} - \mathcal{P}^k \rangle. \tag{2.29}$$

The Kullback-Leibler divergence $\mathrm{KL}(\cdot \| \cdot)$ is a natural choice for D when optimizing over probability distributions. This leads to the proximal iteration:

$$\mathcal{P}_{k+1} = \arg\min_{\mathcal{P} \in \hat{\Pi}(\rho_0, \rho_T)} \; G(\mathcal{P}) + \frac{1}{\eta} \mathrm{KL}(\mathcal{P} \| \mathcal{P}_k) + \langle \frac{\delta F}{\delta \mathcal{P}}(\mathcal{P}_k), \mathcal{P} \rangle. \tag{2.30}$$

At each proximal step, we approximate the function F to second order and the dynamics to first order around the current iterate. This yields a linear covariance steering subproblem that admits a closed-form solution (Yu, Z. Chen, and Y. Chen, 2023).

Iterative Linear Approximation

For nonlinear control-affine systems $dX_t = f(t, X_t)dt + B(t)(u_t dt + \sqrt{\epsilon} dW_t)$, we linearize around the current nominal trajectory (z_t^k, Σ_t^k) at each iteration k:

$$A_t^k = \left.\frac{\partial f}{\partial X}\right|_{z_t^k}, \quad a_t^k = f(z_t^k) - A_t^k z_t^k. \tag{2.31}$$

The linearized system $dX_t = (A_t^k X_t + a_t^k + B_t u_t)dt + B_t dW_t$ defines a linear-Gaussian subproblem to which proximal covariance steering can be applied. The outer loop updates the nominal:

$$(z_t^{k+1}, \Sigma_t^{k+1}) \leftarrow \text{solve linear covariance steering on } (A_t^k, a_t^k, B_t). \tag{2.32}$$

This iterative scheme resembles classical differential dynamic programming (DDP) or iterative LQG, but operates directly on trajectory distributions rather than nominal trajectories. The proximal framework guarantees monotonic improvement in the original nonlinear objective, with convergence rate depending on the linearization quality.

Comparison: GVIMP vs. Proximal Covariance Steering

Figure 2.3 illustrates the comparison between GVIMP (GVI-MP) and Proximal Covariance Steering (PCS-MP) on a 2-link planar arm. Both methods produce distributional solutions, but with complementary characteristics:

- **GVIMP** maximizes entropy subject to cost constraints, yielding broader distributions that naturally promote robustness. It requires no covariance boundary conditions beyond the prior.
- **PCS-MP** steers to prescribed terminal covariances, providing precise uncertainty control at endpoints. It is computationally more efficient (Table 2.2) but requires specification of desired terminal uncertainty.
- Both methods significantly outperform deterministic GPMP2 in robustness metrics (hardware experiments, Section 2.7).

The proximal covariance steering perspective is developed more fully in Chapter 2.6, where closed-form solutions for linear systems and convergence analysis for the iterative nonlinear algorithm are presented. The key insight is that both

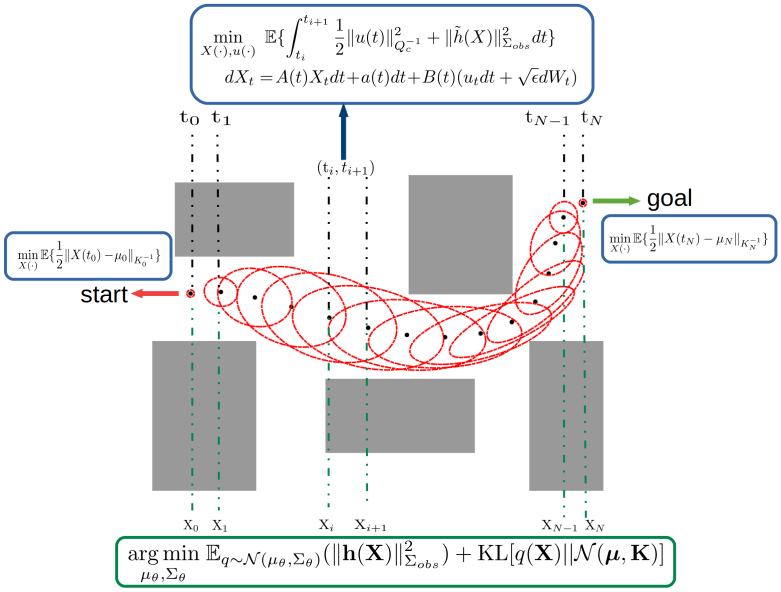


Figure 2.2. Visual demonstration of the control-inference duality underlying both GVIMP and PCS-MP. Left: Stochastic optimal control formulation with explicit dynamics and costs. Right: Equivalent probabilistic inference formulation where planning becomes posterior inference under a cost-augmented prior.

GVIMP and PCS-MP solve related but distinct problems: GVIMP finds the best Gaussian approximation to a posterior, while PCS-MP finds the minimum-energy path distribution satisfying boundary covariance constraints. These complementary perspectives provide flexibility in addressing different planning requirements.

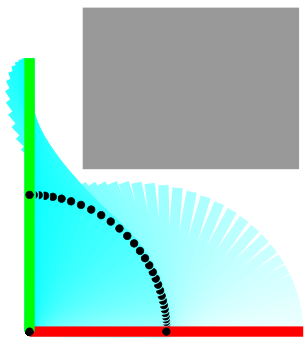
2.7 Experimental Validation

We validate GVIMP through three sets of experiments: (1) point robots in cluttered 2D environments, (2) high-dimensional manipulator arms in workspace obstacles, and (3) hardware experiments demonstrating robustness to perception uncertainties. For collision checking, robots are modeled as spheres of radius r placed at designated body points, and signed distance fields (SDFs) evaluate obstacle proximity along the trajectory.

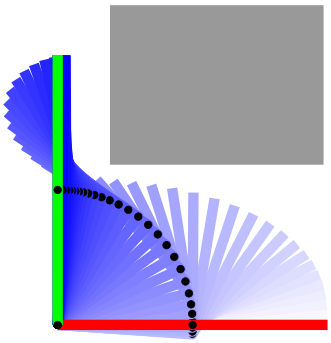
Point Robot Navigation

We evaluate a 2-DOF point robot with state $X_t = [p_x, p_y, v_x, v_y]^T$ navigating cluttered environments (Figure 2.4). Four different start-goal pairs are tested, each requiring navigation through narrow passages between obstacles.

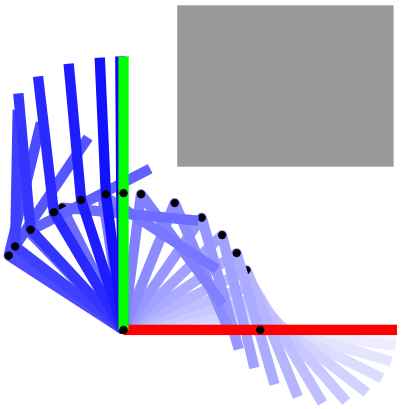
GVIMP successfully finds collision-free trajectory distributions in all four cases. The algorithm is run with temperature scheduling: first solving under low temper-



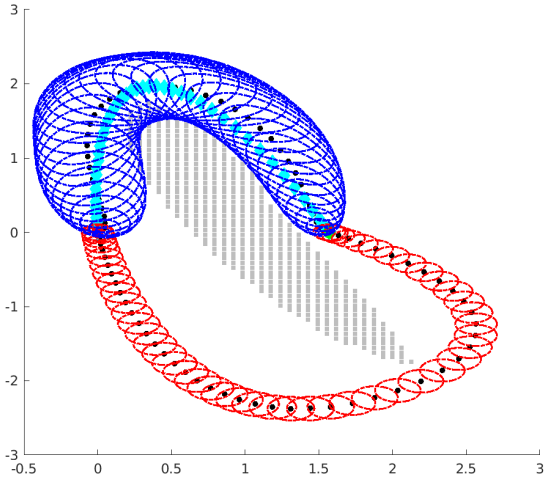
(a) GPMP2 (deterministic)



(b) GVI-MP (max-entropy)



(c) PCS-MP (covariance steering)



(d) Configuration space comparison

Figure 2.3. Comparison of motion planning approaches on a 2-link arm. (a) GPMP2: deterministic MAP trajectory. (b) GVI-MP: maximum-entropy distributional solution with broader uncertainty. (c) PCS-MP: covariance-steered solution with prescribed terminal uncertainty. (d) Configuration space showing GPMP2 path (cyan) and 3σ uncertainty ellipses for GVI-MP (blue) and PCS-MP (red).

ature to obtain collision-free paths, then increasing temperature to promote entropy and robustness. The resulting distributions (Figure 2.4a) are broader than those from covariance steering methods (Figure 2.4b), reflecting the explicit entropy maximization that encourages robustness to modeling and sensing uncertainties.

High-Dimensional Manipulator Planning

2-Link Planar Arm. We compare GPMP2, GVI-MP, and PCS-MP on a 2-DOF planar arm with state $X_t = [q_1, q_2, \dot{q}_1, \dot{q}_2]^T$ performing obstacle avoidance (Figure 2.3). In configuration space (Figure 2.3d), GPMP2's deterministic path is shown in cyan, while GVI-MP (blue) and PCS-MP (red) depict 3σ covariance contours.

Both GVI-MP and PCS-MP provide distributional solutions with quantified uncertainty, while maintaining collision-free mean trajectories. The ellipsoidal uncertainty regions adapt to the local obstacle geometry, being tighter in constrained regions and broader in open space.

7-DOF WAM Arm. We test two bookshelf navigation tasks on a 7-DOF WAM arm (Rooks, 2006) using ROS (Quigley et al., 2009) and MoveIt (Coleman et al., 2014). The state consists of 7D joint position and velocity ($n = 14$). In both tasks (Figures 2.5 and 2.6), the end-effector starts on a table and moves to a goal inside the bookshelf while avoiding shelf collisions.

All three methods (GPMP2, GVI-MP, PCS-MP) successfully plan collision-free trajectories through the constrained workspace. The distributional methods (GVI-MP and PCS-MP) additionally provide uncertainty quantification, enabling downstream robustness analysis.

Computational Performance Analysis

Importance of Closed-Form Prior. Table 2.1 demonstrates the critical importance of the closed-form prior factor computation (Lemma 1). For the 7-DOF arm, the marginal $q_{i,i+1}$ is 28-dimensional. A 3-degree full-grid Gauss-Hermite quadrature would require $3^{28} \approx 10^{13}$ points (computationally infeasible), while even sparse-grid quadrature needs $28^3 = 21{,}952$ points per expectation. The closed-form expression avoids this entirely, reducing computation time from intractable to 0.31 seconds.

	Closed-form	Sparse-grid GH	Full-grid GH
2-DOF Arm	**0.0150 s**	0.0180 s	0.0954 s
7-DOF WAM	**0.3086 s**	–	∞

Table 2.1. Time to compute prior costs in GVI-MP ($N = 50$ states; averaged over 50 runs). For 7-DOF, full-grid quadrature is computationally infeasible; sparse-grid is omitted as it would require 28^3 evaluations per iteration.

End-to-End Runtime. Table 2.2 compares total optimization time on the WAM tasks (averaged over 50 runs). GPMP2 is the fastest as a deterministic baseline (~0.2s). GVI-MP takes significantly longer (14-25s) due to optimizing over the full joint distribution: for 7-DOF with N=50, GPMP2 has 700 variables while GVI-MP has 10,500 variables (mean and covariance). Despite leveraging factorization and closed-form prior updates, GVI-MP's computational cost remains higher than GPMP2.

PCS-MP (Chapter 2.6) is much faster (~0.6s), approaching GPMP2's efficiency. This speedup comes from: (1) parameterizing via the LTV-SDE (A_t, a_t, B_t) rather than joint distribution, and (2) using quadratic approximations instead of Gauss-Hermite quadrature. However, PCS-MP requires additional covariance boundary conditions, whereas GVI-MP is more flexible.

	GPMP2	GVI-MP	PCS-MP
WAM Task 1	**0.20 s**	14.40 s	0.63 s
WAM Task 2	**0.17 s**	24.59 s	0.55 s

Table 2.2. Optimization time for 7-DOF WAM experiments (averaged over 50 runs, $N = 50$).

The computational bottleneck in GVI-MP is addressed in Chapter 3 through GPU parallelization and proximal algorithms, achieving over 97% speedup while maintaining distributional solutions.

Robustness to Uncertainties

Trajectory Resampling. resampling

	GPMP2	GVI-MP	PCS-MP
WAM Task 1	–	0.009 s	0.003 s
WAM Task 2	–	0.003 s	0.002 s

Table 2.3. Sampling time for drawing 1000 trajectories from the optimized distribution (7-DOF WAM, averaged over 50 runs). GPMP2 has no distribution to sample from.

Hardware Validation: Franka Arm. We evaluate robustness on a real Franka Panda arm under perception noise and obstacle displacements (G. Chen et al., 2022). The robot moves from a start pose to a goal while avoiding box obstacles (Figure 2.8). Two experimental scenarios are tested: (1) a single convex box, and (2) a non-convex obstacle configuration.

The experimental procedure introduces two sources of uncertainty:

1. **Perception noise:** Obstacle poses are estimated by averaging noisy camera measurements.

2. **Position disturbances:** Obstacles are manually displaced before execution to simulate real-world variations.

Plans are computed using nominal (averaged) obstacle positions, then executed without replanning after perturbations. During execution, trajectories are resampled from the optimized distribution when needed.

Over 50 randomly disturbed configurations per scenario, we compute the minimum signed distance from the trajectory to actual obstacles. Results appear in Table 2.4 and Figure 2.9. GVIMP achieves the largest average minimum distance (5.46cm in Scenario 1, 0.81cm in Scenario 2), remaining collision-free across all trials. In contrast, baseline planners (RRT-Connect, RRT*, CHOMP, STOMP) exhibit negative distances (collisions) in multiple trials.

Planner	Scenario 1 (cm)	Scenario 2 (cm)
PRM	1.85	−2.84
RRT	−0.11	−3.75
RRT-Connect	−2.37	−1.94
RRT*	1.11	−3.01
RRT-Connect+CHOMP	0.91	−3.28
RRT-Connect+STOMP	−5.98	−0.65
GVIMP	**5.46**	**0.81**

Table 2.4. Average minimum signed distance to disturbed obstacles over 50 trials. Positive values indicate collision-free execution; negative values indicate penetration. GVIMP demonstrates superior robustness.

These hardware results validate GVIMP's robustness benefits in real-world conditions. By reasoning about trajectory distributions rather than nominal paths, GVIMP naturally maintains safety margins that accommodate perception errors and environmental variations.

Planar Quadrotor Example

We conclude with a planning example for a linearized planar quadrotor, validating GVIMP on continuous-state aerial systems. The 6D state is $X_t = [p_x, p_z, \phi, v_x, v_z, \dot{\phi}]^T$, where (p_x, p_z) are position, (v_x, v_z) are body-frame velocities, and ϕ is roll angle.

The nonlinear dynamics are:

$$\dot{X}_t = \begin{bmatrix} v_x \cos\phi - v_z \sin\phi \\ v_x \sin\phi + v_z \cos\phi \\ \dot{\phi} \\ v_z\dot{\phi} - g\sin\phi \\ -v_x\dot{\phi} - g\cos\phi \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1/m & 1/m \\ l/J & -l/J \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \tag{2.33}$$

with $m = 1/\sqrt{2}$, $l = \sqrt{2}$, $J = 1$, and thrust inputs u_1, u_2.

We iteratively linearize (2.33) around the current nominal trajectory, apply GVIMP to the resulting LTV system, and update the nominal. Figure 2.10 shows results for four planning tasks with $N = 3000$ support states (enabling fine temporal resolution). GVIMP successfully navigates the quadrotor through cluttered environments while maintaining quantified uncertainty bounds throughout the trajectory.

The covariance evolution plots (Figures 2.10e–2.10f) reveal how uncertainty propagates through the nonlinear dynamics. Velocity covariances increase during aggressive maneuvers and decrease during straight-line flight, while angular covariances reflect the coupling between roll angle and linear motion in the quadrotor model. These results demonstrate GVIMP's applicability to continuous-state systems with coupled dynamics.

2.8 Chapter Summary

This chapter introduced the Gaussian Variational Inference Motion Planning (GVIMP) framework for probabilistic motion planning under uncertainty. The key contributions and insights include:

Planning-as-Inference Duality. By formulating motion planning as variational inference, GVIMP bridges stochastic optimal control and probabilistic reasoning. The optimal trajectory distribution solving a stochastic control problem can be characterized as a posterior distribution that balances prior dynamics (smooth motion) against task costs (collision avoidance, goal reaching). Approximating this posterior within the Gaussian family yields tractable optimization while maintaining uncertainty quantification.

Natural Gradient Descent with Sparse Factorization. The natural gradient algorithm accounts for the Riemannian geometry of distribution space, providing

faster convergence than Euclidean gradients. By exploiting the sparse factor graph structure underlying motion planning—where dynamics couple only adjacent time steps and costs factorize across states—we decompose the gradient computation into independent marginal expectations. This factorization enables efficient implementation (Chapter 3) and theoretical analysis.

Closed-Form Prior Updates. For linear-Gaussian dynamics, the prior factors admit closed-form gradient computation (Lemma 1), avoiding expensive numerical quadrature. This hybrid approach—combining analytic prior updates with sparse quadrature for collision factors—balances accuracy and efficiency, making GVIMP practical for high-dimensional systems.

Proximal Gradient Perspective in Distribution Space. Section 2.6 presented an alternative formulation based on proximal gradients in the space of path distributions. By connecting GVIMP to optimal transport theory and covariance steering, this perspective reveals the relationship between maximum-entropy planning and minimum-energy path control. The iterative linearization scheme for nonlinear systems provides a practical algorithm with guaranteed monotonic improvement. This dual perspective—variational inference versus distributional control—enriches our understanding of probabilistic motion planning.

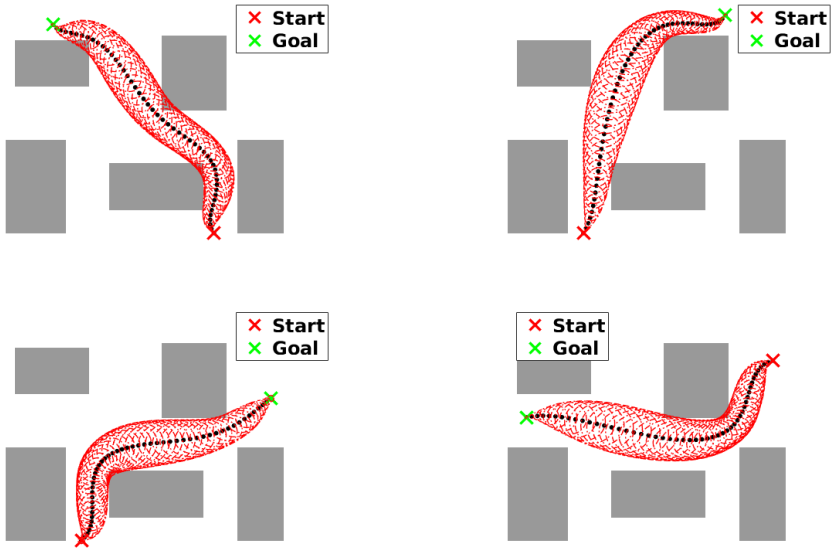
Entropy Regularization and Robustness. The entropy term in the GVIMP objective naturally promotes robustness by encouraging distributional diversity. This connection to maximum-entropy planning and reinforcement learning provides a principled mechanism for trading off nominal optimality against resilience to uncertainties. Hardware experiments (Section 2.7) confirm that distributional planning significantly improves robustness compared to deterministic methods.

Experimental Validation. Extensive experiments across point robots, manipulator arms, and aerial vehicles demonstrate GVIMP's effectiveness. Hardware trials on a Franka arm show superior robustness to perception noise and obstacle displacements, with GVIMP maintaining collision-free execution across 50 random perturbations while baseline methods collide. The ability to rapidly resample from the optimized distribution enables reactive adaptation without full replanning. Comparisons with Proximal Covariance Steering (PCS-MP) reveal complementary

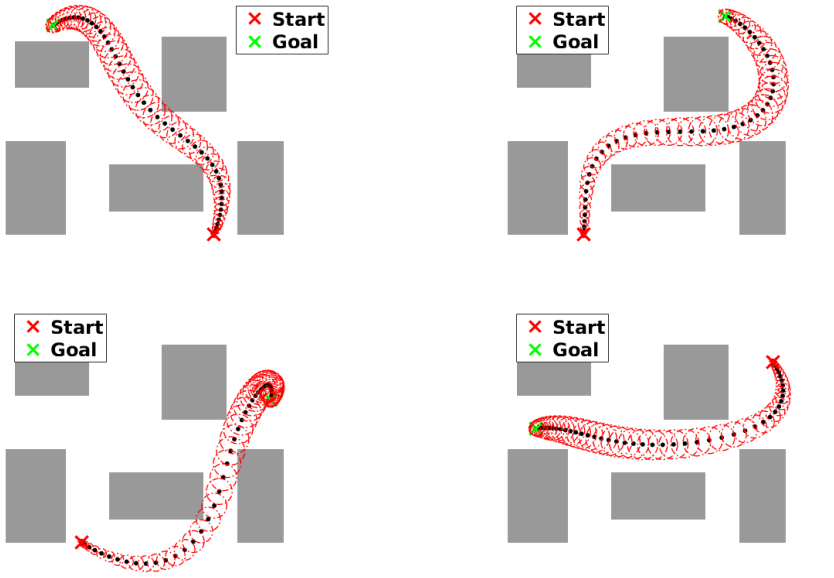
trade-offs: GVIMP maximizes entropy for robustness, while PCS-MP achieves precise terminal covariance control with greater computational efficiency.

Connections to Subsequent Chapters. The computational bottleneck identified in GVIMP—$\mathcal{O}(Nn^{k_q})$ complexity from collision factor evaluation—motivates Chapter 3, which achieves over 97% speedup through GPU parallelization and proximal algorithms. The covariance steering perspective introduced in Section 2.6 is developed more fully in Chapter 2.6, which provides closed-form solutions for linear systems and convergence analysis. The hybrid system extension (Chapter 4) addresses systems with discrete mode switches, complementing GVIMP's continuous-flow focus.

In summary, GVIMP provides a principled, computationally tractable approach to probabilistic motion planning that explicitly balances optimality, safety, and robustness. The framework's flexibility—accommodating various dynamics models, cost structures, and uncertainty representations through both variational inference and distributional control perspectives—makes it broadly applicable to robotic decision-making under uncertainty.

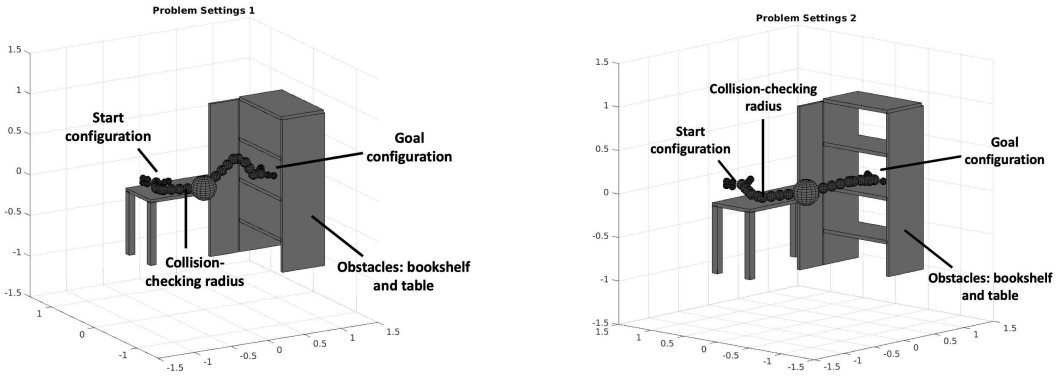


(a) GVI-MP with entropy regularization



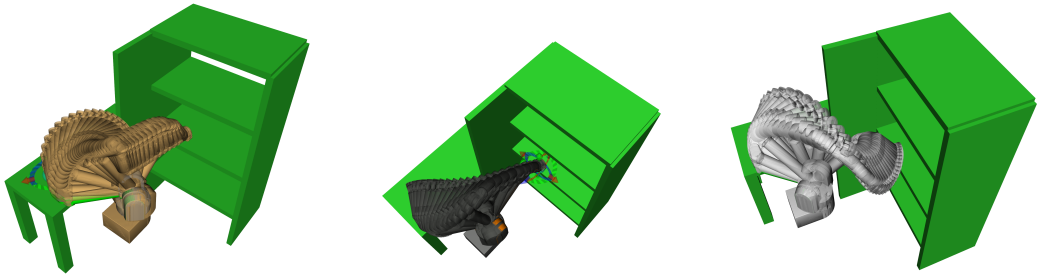
(b) PCS-MP (Chapter 2.6)

Figure 2.4. 2D point-robot planning with $N = 50$ support states. GVI-MP yields broader distributions reflecting robustness to uncertainties, while PCS-MP achieves tighter covariance control.

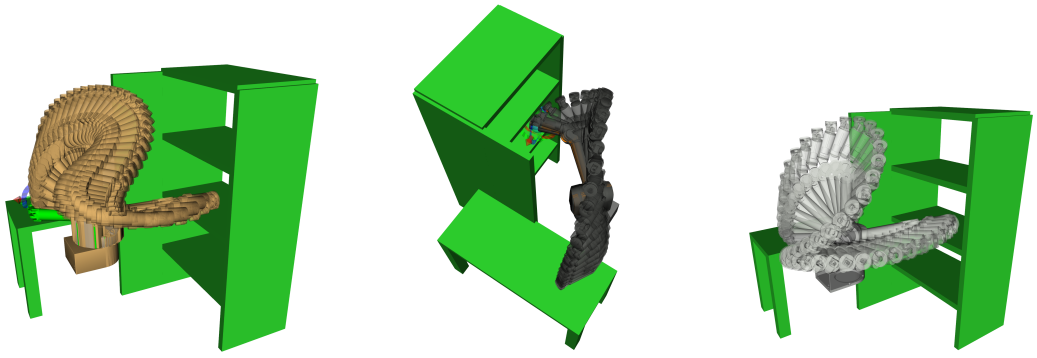


(a) Task 1 **(b)** Task 2

Figure 2.5. WAM arm problem settings. Black spheres denote collision-checking points with safety radius r.



(a) GPMP2, Task 1 **(b)** GVI-MP, Task 1 **(c)** PCS-MP, Task 1



(d) GPMP2, Task 2 **(e)** GVI-MP, Task 2 **(f)** PCS-MP, Task 2

Figure 2.6. 7-DOF WAM arm planning visualized in ROS MoveIt. Top row: Task 1. Bottom row: Task 2. All methods successfully navigate the bookshelf while maintaining collision-free trajectories.

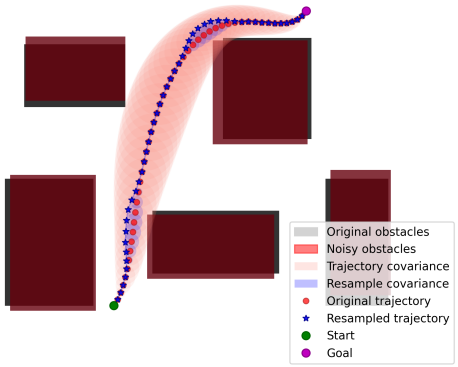
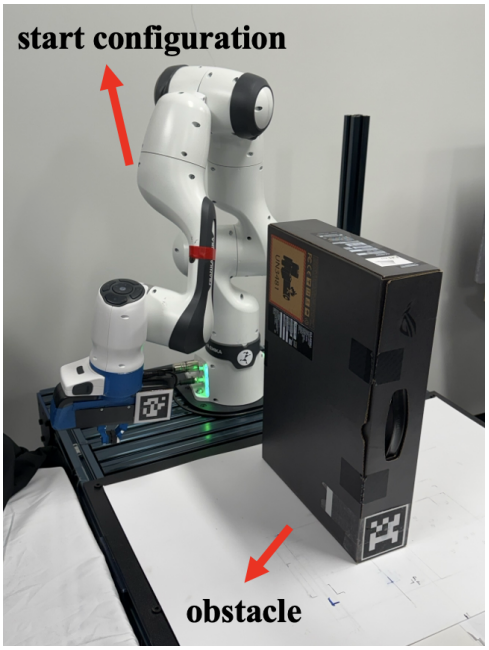
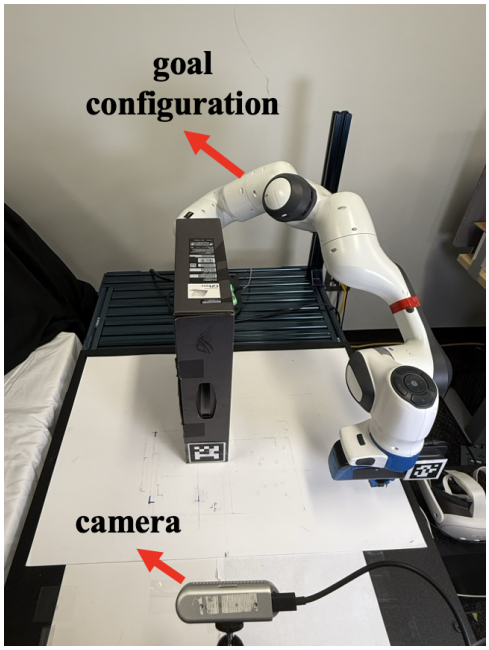


Figure 2.7. 2D point-robot resampling under obstacle perturbations. Original plan (blue) and resampled trajectories (red) from the optimized covariance. Resampling enables reactive adaptation without full replanning.

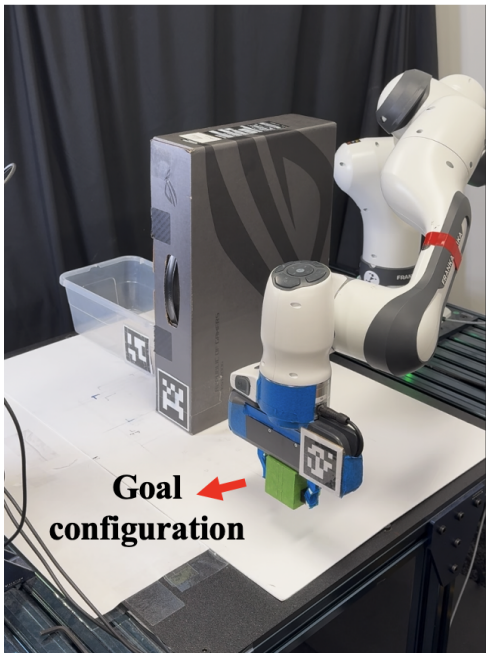


(a) Scenario 1: Start



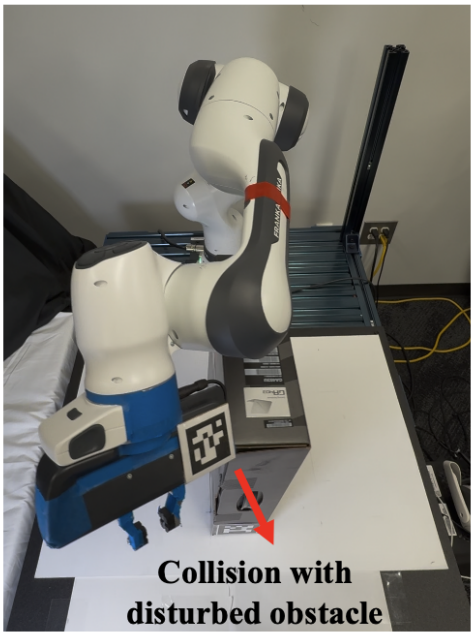
(b) Scenario 1: Goal

(c) Scenario 2: Start



(d) Scenario 2: Goal

Figure 2.8. Franka hardware experiment settings. Scenario 1 features a single box obstacle. Scenario 2 includes additional non-convex constraints near the start configuration.



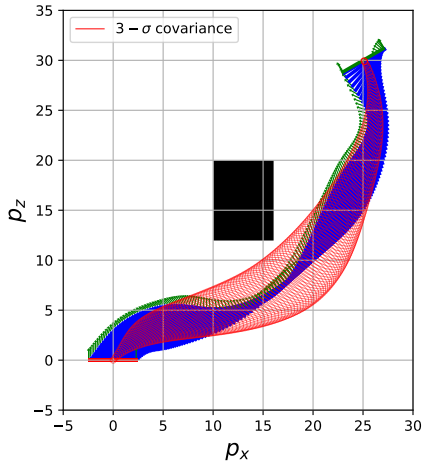
(a) RRT-Connect (collision)

(b) GVIMP (collision-free)

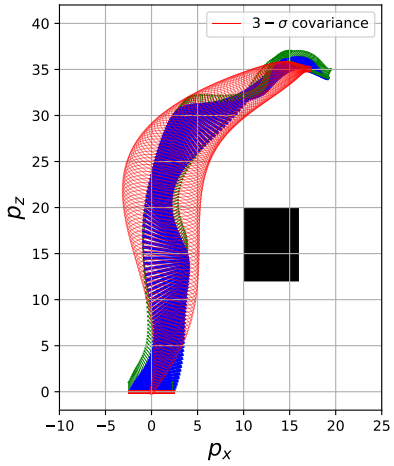
(c) RRT-Connect+STOMP (collision)

(d) GVIMP (collision-free)

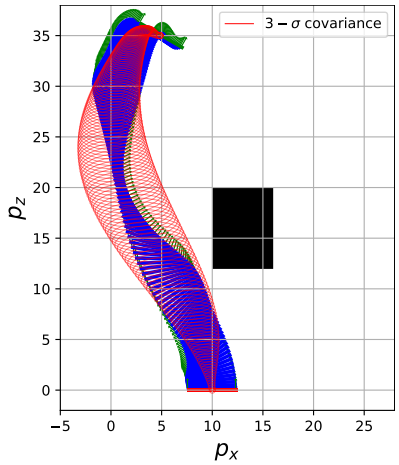
Figure 2.9. Hardware execution under disturbed obstacles (one representative trial per scenario). Top: Scenario 1. Bottom: Scenario 2. Baseline planners collide with perturbed obstacles; GVIMP remains safe through distributional planning and adaptive resampling. Additional results: `https://www.youtube.com/watch?v=c4sFOlEki0Q`



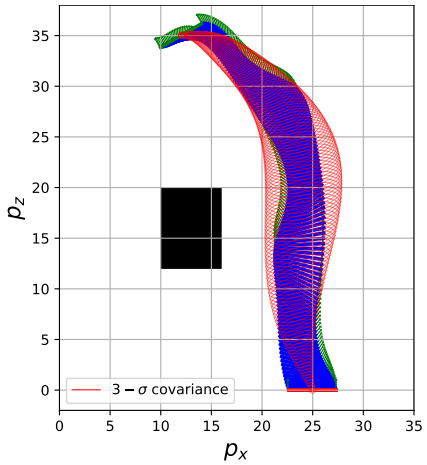
(a) Task 1



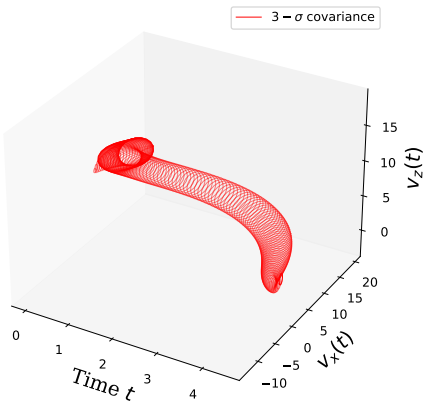
(b) Task 2



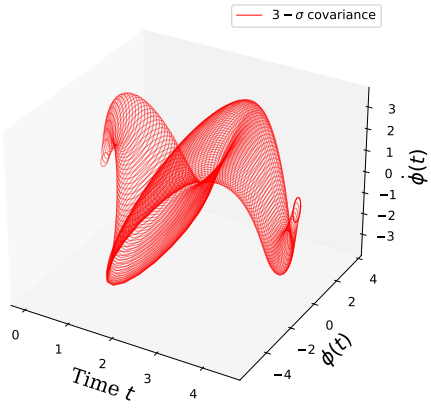
(c) Task 3



(d) Task 4



(e) Cov$[v_x(t), v_z(t)]$ for Task 2



(f) Cov$[\phi(t), \dot{\phi}(t)]$ for Task 2

Figure 2.10. Planar quadrotor planning via iterative linearization and GVIMP ($N = 3000$ states). Top: Four planning scenarios showing mean paths (solid lines) and 3σ position uncertainty (red ellipses). Bottom: Covariance evolution for velocity and angular state in Task 2, demonstrating uncertainty propagation through the dynamics.

Chapter 3

PARALLEL GAUSSIAN VARIATIONAL INFERENCE MOTION PLANNING

The Gaussian Variational Inference Motion Planning (GVIMP) framework introduced in Chapter 1 provides a principled approach to motion planning under uncertainty by formulating the problem as variational inference over trajectory distributions. However, the computational complexity of GVIMP, particularly when evaluating gradients over dense trajectory representations, presents a significant bottleneck for real-time applications. This chapter addresses these computational challenges through two key innovations: (1) a proximal point algorithm that exploits the splitting structure of the motion planning objective, and (2) parallel computation strategies leveraging GPU architectures and sparse factor graph structures.

We first derive a Kullback-Leibler (KL) proximal algorithm for linear-Gaussian stochastic systems, which provides stable iterative updates with convergence guarantees. The sparse structure of the underlying planning factor graph is then exploited to distribute gradient computations across marginal distributions, enabling efficient parallel execution on GPUs. For nonlinear systems, we propose an iterative paradigm based on Statistical Linear Regression (SLR), where the parallel algorithm serves as an efficient subroutine at each linearization step. The resulting framework, termed *Parallel Gaussian Variational Inference Motion Planning (P-GVIMP)*, achieves significant computational speedup while maintaining the theoretical consistency of the original GVIMP formulation.

3.1 KL-Proximal Variational Inference for Motion Planning

The GVIMP formulation seeks to minimize the KL divergence between a proposal Gaussian distribution $q_\theta \sim \mathcal{N}(\mu_\theta, \Sigma_\theta)$ and the target posterior $\tilde{q}^\star$ induced by the stochastic control problem. For linear time-varying (LTV) systems of the form

$$dX_t = (A_t X_t + a_t + B_t u_t)dt + B_t dW_t, \tag{3.1}$$

the discrete-time variational objective becomes

$$\mathcal{J}(q_\theta) = \mathbb{E}_{q_\theta}[\psi(\mathbf{x})] - \hat{T}\mathcal{H}(q_\theta), \tag{3.2}$$

where $\psi(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \boldsymbol{\mu}\|^2_{\mathbf{K}^{-1}} + V(\mathbf{x})$ encodes the prior dynamics and collision costs, and $\hat{T}$ is a temperature parameter controlling the entropy regularization.

Proximal Point Algorithm with Splitting

The proximal point algorithm in distribution space replaces the Euclidean regularizer with the KL divergence, yielding the iteration

$$\theta_{k+1} = \arg\min_{\theta} \mathcal{J}(\theta) + \frac{1}{\beta_k} \mathrm{K}L\left(q_\theta \parallel q_{\theta_k}\right), \tag{3.3}$$

where β_k is the step size. The key observation is that the objective $\mathcal{J}$ admits a natural splitting into two parts:

$$\psi_e(\theta) \triangleq \mathbb{E}_{q_\theta}\left[\log q_\theta + \frac{1}{2}\|\mathbf{x} - \boldsymbol{\mu}\|^2_{\mathbf{K}^{-1}}\right], \tag{3.4a}$$

$$\psi_d(\theta) \triangleq \mathbb{E}_{q_\theta}\left[V(\mathbf{x})\right]. \tag{3.4b}$$

The term ψ_e corresponds to the negative log-likelihood of the prior and entropy, which forms a conjugate model with closed-form gradients. The term ψ_d represents the collision cost, which is non-conjugate and requires approximation.

By approximating ψ_d to first order around θ_k, we obtain explicit update rules that only require evaluating $\nabla_{\mu_\theta}\psi_d(\theta_k)$ and $\nabla_{\Sigma_\theta}\psi_d(\theta_k)$ at the current iterate.

Theorem 1 (KL-Proximal Updates)**.** *The KL-proximal updates for the mean and precision are given by*

$$(\mathbf{K}^{-1} + \frac{\Sigma^{-1}_{\theta_k}}{\beta_k})\mu_{\theta_{k+1}} = -\nabla_{\mu_\theta}\psi_d(\theta_k) + \mathbf{K}^{-1}\boldsymbol{\mu} + \frac{\Sigma^{-1}_{\theta_k}}{\beta_k}\mu_{\theta_k}, \tag{3.5a}$$

$$\Sigma^{-1}_{\theta_{k+1}} = \frac{\beta_k}{\beta_k + 1}\left(2\nabla_{\Sigma_\theta}\psi_d(\theta_k) + \mathbf{K}^{-1} + \frac{\Sigma^{-1}_{\theta_k}}{\beta_k}\right). \tag{3.5b}$$

Under standard smoothness and convexity assumptions, the algorithm converges linearly with step size $0 < \beta_k < \alpha/L$, where L is the smoothness constant of ψ_d and α measures the strong monotonicity of the KL divergence. The first-order approximation error is bounded by $O(\|\theta - \theta_k\|^2)$.

Connection to Natural Gradient Descent

The proposed KL-proximal algorithm is closely related to natural gradient descent (NGD). NGD with unit step size solves

$$\delta\theta = \arg\min_{d\theta} \; d\theta^T \nabla_\theta \mathcal{L}(\theta) + \mathrm{K}L^{sym}(q_\theta \parallel q_{\theta + d\theta}), \tag{3.6}$$

where $\mathrm{K}L^{sym}$ denotes the symmetric KL divergence. Our approach instead uses the forward KL divergence $\mathrm{K}L(q_{\theta_{k+1}} \| q_{\theta_k})$ as the proximal term, which offers superior stability and admits explicit updates via the splitting structure.

3.2 Distributed Computation on Sparse Factor Graphs

The primary computational bottleneck in evaluating the updates (3.5) lies in computing the gradients $\nabla_{\mu_\theta}\psi_d$ and $\nabla_{\Sigma_\theta}\psi_d$. By Stein's Lemma, these gradients can be expressed as

$$\nabla_{\mu_\theta}\psi_d(\theta) = \mathbb{E}_{q_\theta}\left[\nabla_{\mathbf{x}}V(\mathbf{x})\right], \tag{3.7a}$$

$$\nabla_{\Sigma_\theta}\psi_d(\theta) = \frac{1}{2}\mathbb{E}_{q_\theta}\left[\nabla^2_{\mathbf{xx}}V(\mathbf{x})\right]. \tag{3.7b}$$

Since the collision cost factorizes as $V(\mathbf{x}) = \sum_{i=0}^{N} V_i(x_i)$, where $x_i = M_i\mathbf{x}$ are marginal states, the gradients decompose into

$$\nabla_{\mu_\theta}\psi_d = \sum_{i=0}^{N} M_i^T \nabla_{\mu_\theta^i}\psi_i^d, \tag{3.8a}$$

$$\nabla_{\Sigma_\theta}\psi_d = \sum_{i=0}^{N} M_i^T \nabla_{\Sigma_\theta^i}\psi_i^d \, M_i, \tag{3.8b}$$

where $q_i \sim \mathcal{N}(\mu_\theta^i, \Sigma_\theta^i)$ denotes the marginal distribution at time i.

Gaussian Belief Propagation for Marginals

Computing the marginal covariances Σ_θ^i directly from the joint precision matrix Σ_θ^{-1} requires a costly matrix inversion with $O((Nn)^3)$ complexity. Instead, we leverage the sparse structure of the precision matrix, which for linear Gaussian processes has a tridiagonal block structure. Gaussian Belief Propagation (GBP) (Shental et al., 2008) computes marginals via local message passing on the factor graph in $O(Nn^3)$ time.

For a Gaussian $q(\mathbf{x}) \propto \exp(-\frac{1}{2}\mathbf{x}^T\Lambda_\theta\mathbf{x})$ with sparse precision Λ_θ, GBP iteratively computes messages between variables and factors:

$$m_{x_i \to f_j} = \prod_{s \in \mathcal{N}(i)\setminus j} m_{f_s \to x_i}, \tag{3.9a}$$

$$m_{f_j \to x_i} = \sum_{X_j \setminus x_i} f_j(X_j) \prod_{k \in \mathcal{N}(j)\setminus i} m_{x_k \to f_j}, \tag{3.9b}$$

where $\mathcal{N}(i)$ denotes neighbors in the factor graph. The marginal precision at node i is then

$$\Lambda_i = \sum_{s \in \mathcal{N}(i)} \Lambda_{f_s \to x_i}. \tag{3.10}$$

This reduces the computational bottleneck from matrix inversion to sparse linear operations.

Parallel Collision Checking on GPU

The factorized gradient computation (3.8) naturally parallelizes across the N marginal factors. On each marginal, the expectation is approximated using sparse Gauss-Hermite quadrature with complexity $O(n^{k_q})$, where k_q is the desired polynomial precision (Heiss and Winschel, 2008).

For a dense trajectory with $N \gg n$, the serial complexity is $O(Nn^{k_q})$, which becomes prohibitive for high-dimensional systems or fine discretizations. By distributing the computation across GPU cores, each factor is evaluated independently, reducing the wall-clock time to $O(n^{k_q})$ plus communication overhead. Combined with GBP for marginal computation, the total algorithm complexity becomes $O(Nn^3)$, dominated by the marginal covariance updates rather than collision checking.

3.3 Iterative P-GVIMP for Nonlinear Systems

The proximal algorithm derived in Section 3.1 applies to linear time-varying (LTV) systems. For nonlinear stochastic dynamics

$$dX_t = f(X_t, u_t)\, dt + g_t(X_t)\, dW_t, \tag{3.11}$$

we adopt the standard Gaussian approximation (Timothy D Barfoot, 2024; Dellaert, 2021) and propose an iterative linearization scheme. At each outer iteration, the nonlinear system is linearized around the current nominal trajectory using Statistical Linear Regression (SLR) (Arasaratnam, Haykin, and Elliott, 2007), yielding an LTV approximation to which P-GVIMP is applied.

Statistical Linear Regression

Given a nominal state distribution $X_i \sim \mathcal{N}(\bar{X}_i, \bar{\Sigma}_i)$ at time t_i, SLR finds the best linear approximation $X_{i+1} \approx A_i X_i + a_i$ in the mean-squared sense:

$$\{A_i, a_i\} = \arg\min \mathbb{E}\left[\|f(X_i) - (A_i X_i + a_i)\|^2\right]. \tag{3.12}$$

The optimal parameters are

$$A_i^* = P_{yx} P_{xx}^{-1}, \tag{3.13a}$$

$$a_i^* = \mathbb{E}[f(X_i)] - A_i^* \mathbb{E}[X_i], \tag{3.13b}$$

where

$$P_{yx} = \mathbb{E}\left[(f(X_i) - \mathbb{E}[f(X_i)])(X_i - \mathbb{E}[X_i])^T\right],$$
$$P_{xx} = \mathbb{E}\left[(X_i - \mathbb{E}[X_i])(X_i - \mathbb{E}[X_i])^T\right].$$

The expectations are computed using sparse Gauss-Hermite quadrature. This linearization is performed sequentially forward in time, yielding an LTV system $\{A_t^k, a_t^k, B_t\}$ at each iteration k.

Algorithm Summary

The complete iterative P-GVIMP algorithm for nonlinear systems is summarized in Algorithm 1.

Algorithm 1: Iterative P-GVIMP for Nonlinear Systems

Input: Nonlinear dynamics (f_t, g_t), initial nominal $(\bar{X}_t^0, \bar{\Sigma}_t^0)$

Output: Optimized trajectory distribution $\mathcal{N}(\mu_\theta^*, \Sigma_\theta^*)$

for $k = 0, 1, 2, \ldots$ *until convergence* **do**

  `// Statistical Linear Regression`

  Linearize system around $(\bar{X}_t^k, \bar{\Sigma}_t^k)$ to obtain (A_t^k, a_t^k, B_t);

  `// Parallel KL-Proximal VI on LTV System`

  for *inner iterations* **do**

    Compute marginals $\{q_i \sim \mathcal{N}(\mu_\theta^i, \Sigma_\theta^i)\}$ via GBP;

    Compute collision factors $\{\nabla_{\mu_\theta^i} \psi_i^d, \nabla_{\Sigma_\theta^i} \psi_i^d\}$ in parallel on GPU;

    Aggregate to joint gradients: $\nabla_{\mu_\theta} \psi_d \leftarrow \sum_i M_i^T \nabla_{\mu_\theta^i} \psi_i^d$;

    Update $(\mu_\theta, \Sigma_\theta^{-1})$ via Theorem 1;

  Update nominal: $(\bar{X}_t^{k+1}, \bar{\Sigma}_t^{k+1}) \leftarrow (\mu_\theta, \Sigma_\theta)$;

The algorithm alternates between linearizing the nonlinear dynamics (outer loop) and solving the linear variational problem (inner loop). Convergence is monitored via the norm difference between consecutive nominal trajectories.

3.4 Experimental Validation

We validate the proposed P-GVIMP framework through three sets of experiments: (1) computational efficiency comparisons demonstrating GPU acceleration, (2) motion planning results for high-dimensional robot arms, and (3) nonlinear planning for a planar quadrotor system.

Computational Efficiency

We evaluate the computational efficiency gains achieved through parallel computation on three representative systems: 2D point robots, 3D point robots, and the 7-DOF WAM arm. All experiments use dense trajectory discretizations with $N = 750$ support states to stress-test the scalability of the approach.

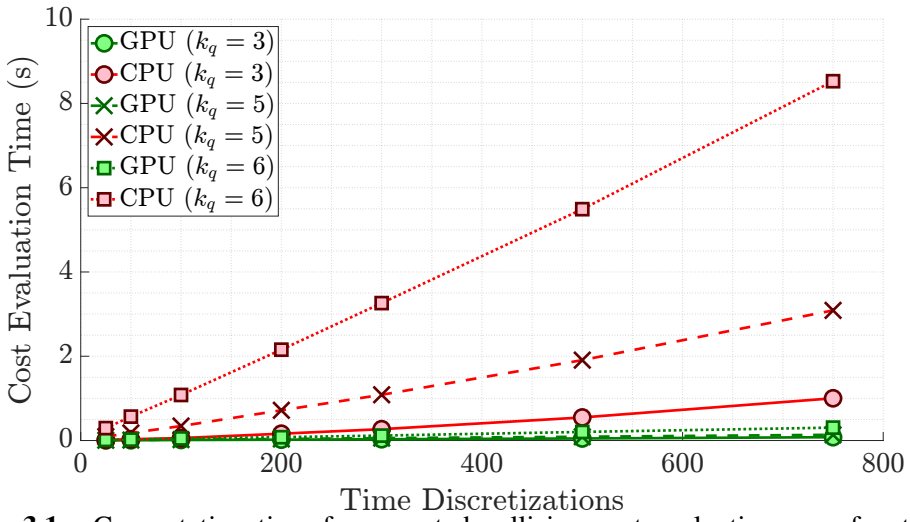


Figure 3.1. Computation time for expected collision cost evaluation as a function of trajectory discretization N for the 7-DOF WAM arm. The parallel GPU implementation (blue) scales much more favorably than the serial CPU implementation (orange).

Parallel Collision Checking. We compare GPU-parallel versus CPU-serial implementations for computing expected collision costs on dense trajectories. Table 3.1 reports the results, showing over 96% speedup for the WAM arm. The efficiency gain increases with system dimensionality, as higher-dimensional state spaces require more quadrature points for accurate expectation estimation.

	2D Robot	3D Robot	7-DoF WAM
Serial	85.40 ms	479.7 ms	8010.5 ms
Parallel	14.80 ms	37.60 ms	306.16 ms
Speedup	82.67%	92.16%	**96.18%**

Table 3.1. Collision checking time comparison (quadrature degree $k_q = 6$ for WAM, $k_q = 10$ for others).

Figure 3.1 shows how computation time scales with the number of discretization points N for the 7-DOF WAM arm. The serial implementation exhibits near-linear growth, while the parallel implementation remains relatively flat due to efficient GPU utilization. The gap widens dramatically as N increases, demonstrating the critical importance of parallel computation for dense trajectory representations.

Gaussian Belief Propagation. Table 3.2 compares GBP against brute-force inversion of the precision matrix. For the 7-DOF system, GBP achieves over 99% speedup, with linear rather than cubic dependence on N. This dramatic improve-

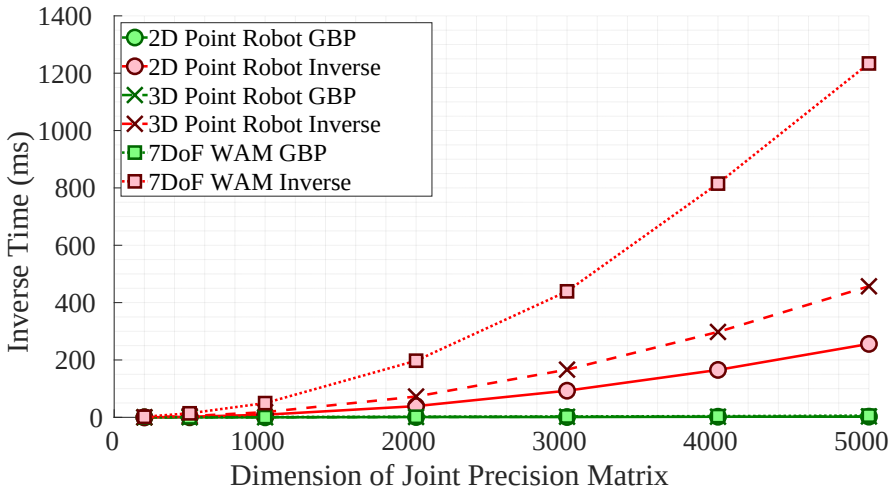


Figure 3.2. Computation time for marginal covariance extraction as a function of precision matrix dimension. Brute-force inversion (orange) scales cubically, while Gaussian Belief Propagation (blue) scales linearly.

ment stems from exploiting the sparse tridiagonal block structure of the precision matrix, which allows local message passing to replace global matrix operations.

	2D Robot	3D Robot	7-DoF WAM
Brute force	256.4 ms	456.86 ms	1234.04 ms
GBP	2.36 ms	3.01 ms	6.04 ms
Speedup	99.07%	99.34%	**99.51%**

Table 3.2. Marginal covariance computation time (precision matrix dimension: 5000).

The scalability of GBP versus direct inversion is illustrated in Figure 3.2. Direct inversion exhibits cubic growth, becoming prohibitive beyond 4000 dimensions, while GBP maintains linear scaling even for very large problems. This efficiency gain is crucial for enabling real-time planning with fine temporal resolution.

Overall Optimization Time. Combining parallel collision checking with GBP-based marginal computation, P-GVIMP achieves 97% speedup for the complete optimization on 7-DOF planning problems compared to the serial GVIMP implementation (Yu and Y. Chen, 2024). Table 3.3 summarizes the end-to-end performance gains.

The computational profile shift is noteworthy: in serial GVIMP, collision checking dominates (85-90% of runtime), while in P-GVIMP, marginal covariance computation becomes the bottleneck (60-70% of runtime). This indicates that further

	2D Robot	3D Robot	7-DoF WAM
Serial GVIMP	7.38 s	27.07 s	425.03 s
P-GVIMP	0.48 s	1.12 s	12.25 s
Speedup	93.50%	95.86%	**97.11%**

Table 3.3. End-to-end optimization time comparison for complete motion planning problems ($N = 750$).

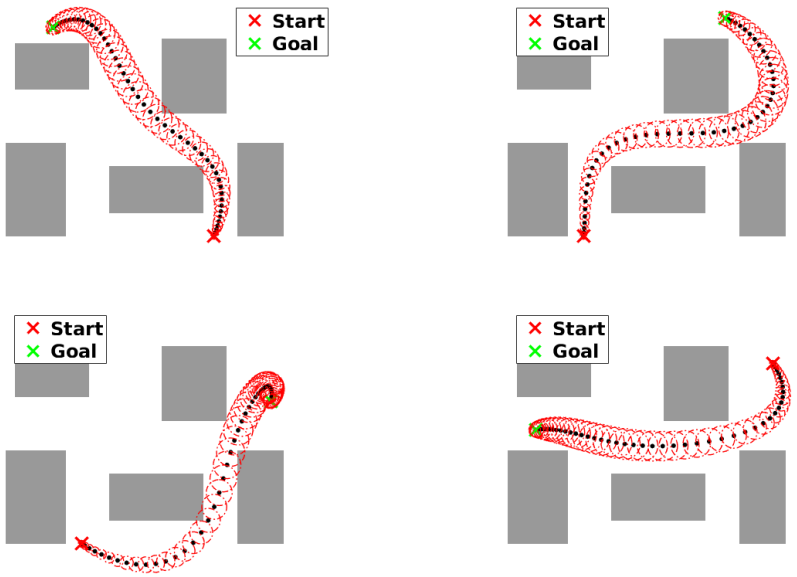
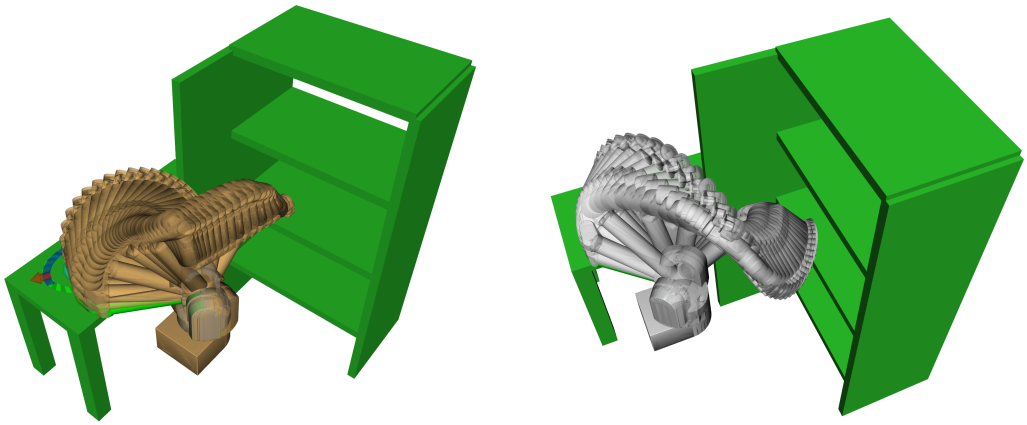


Figure 3.3. P-GVIMP results for 2D point robot planning with four different start-goal configurations ($N = 50$ support states). Mean trajectories are shown as solid lines with 3σ covariance ellipses displayed at selected time steps. The algorithm finds collision-free paths while maintaining quantified uncertainty bounds.

speedups would require parallelizing the GBP message passing, which remains a promising direction for future work.

Visual Results for 2D Planning. To illustrate the planning results, Figure 3.3 shows trajectory distributions obtained for a 2D point robot navigating through a cluttered environment with four different start-goal pairs. The mean trajectories (solid lines) successfully avoid obstacles while the covariance ellipses (shown at selected time steps) quantify the uncertainty along each path. The entropy regularization naturally produces wider distributions in regions with multiple feasible paths, reflecting the algorithm's inherent robustness considerations.



(a) GPMP2 (deterministic) **(b)** P-GVIMP (distributional)

Figure 3.4. WAM arm planning comparison for bookshelf task 1. GPMP2 produces a nominal trajectory (cyan), while P-GVIMP generates a trajectory distribution (gray mean with uncertainty ellipsoids). Both successfully avoid obstacles ($N = 750$ support states).

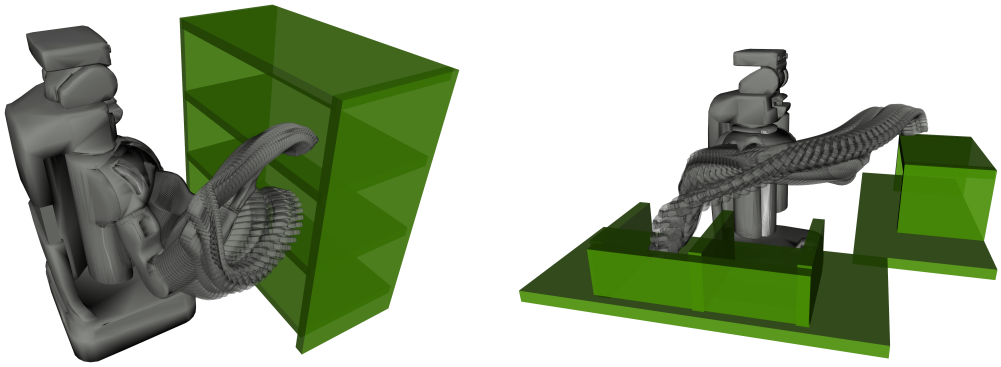
High-Dimensional Robot Arm Planning

We evaluate P-GVIMP on the 7-DOF WAM (Rooks, 2006) and PR2 (Garage, 2012) in cluttered bookshelf environments. These experiments demonstrate the algorithm's capability to handle high-dimensional configuration spaces with complex obstacle geometry.

WAM Arm Results. Figure 3.4 compares trajectories obtained by GPMP2 (deterministic baseline) and P-GVIMP for two bookshelf navigation tasks. While GPMP2 produces a single nominal trajectory, P-GVIMP provides a full trajectory distribution with quantified uncertainty. Both methods successfully avoid collisions, but P-GVIMP's distributional solution enables downstream robustness analysis and resampling capabilities.

PR2 Arm Results. Figure 3.5 shows P-GVIMP results for the PR2 right arm in two different scenarios: a bookshelf environment and an industrial setting with multiple obstacles. The algorithm successfully navigates narrow passages while maintaining safe clearance from obstacles throughout the trajectory.

Robustness to Perception Noise. To assess robustness to perception noise, we computed plans in a nominal environment, then perturbed obstacle positions by adding Gaussian noise and executed the plans without replanning. Figure 3.6 shows trajectories from four different planners in the same perturbed environment. Both



(a) Bookshelf environment **(b)** Industrial environment

Figure 3.5. P-GVIMP results for PR2 right arm in two challenging environments ($N = 750$ support states). Gray trajectories show mean paths with uncertainty visualization.

BIT* and FMT* collide with the displaced obstacles (yellow), while P-GVIMP remains collision-free.

Table 3.4 quantifies the robustness over 50 random obstacle perturbations. P-GVIMP maintains an average minimum distance of 2.16 cm from obstacles, compared to 0.52 cm for GPMP2 and negative distances (collisions) for BIT* and FMT*. The distributional planning approach provides superior safety margins by explicitly accounting for uncertainty in the optimization.

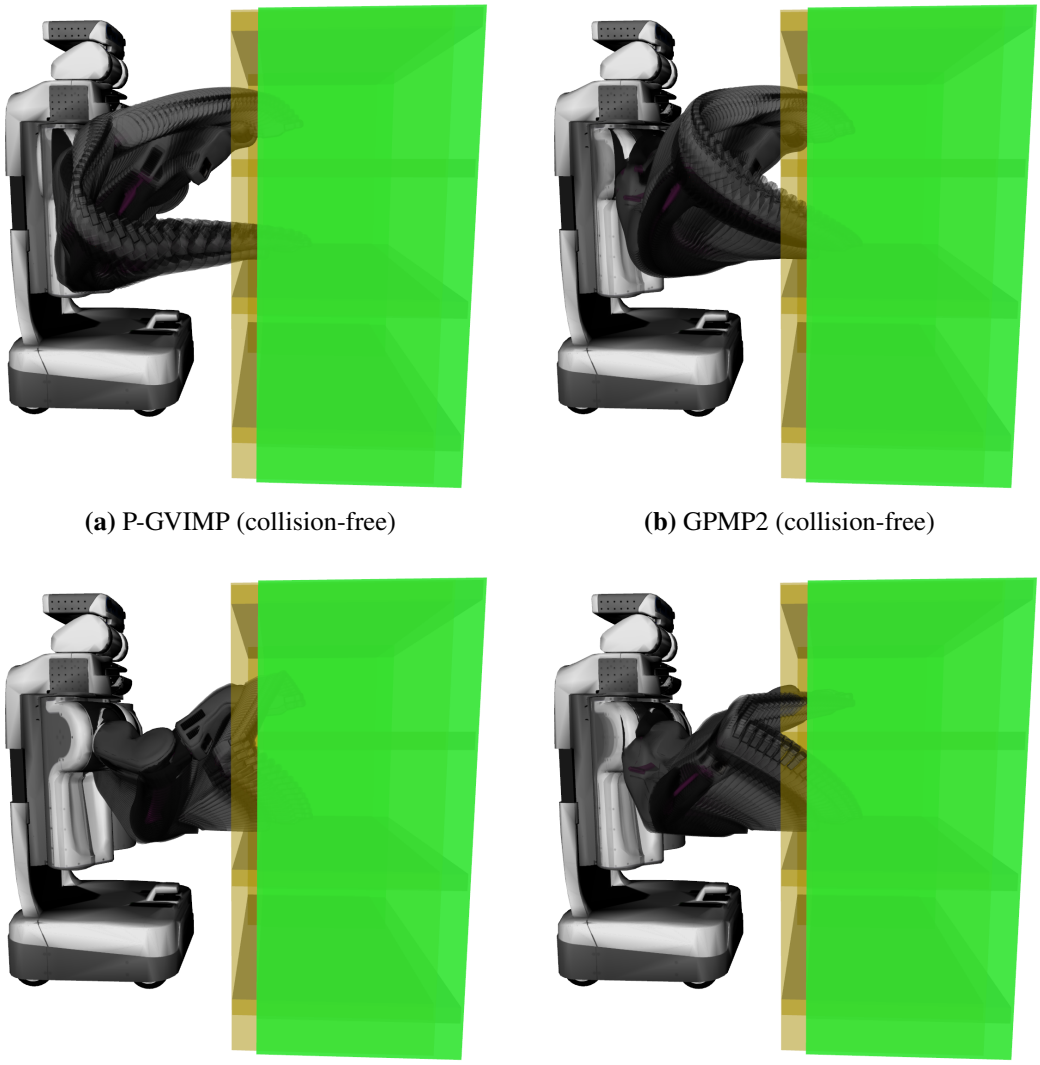
Planner	P-GVIMP	GPMP2	BIT*	FMT*
Avg. distance (cm)	**2.16**	0.52	-3.24	-3.13
Std. dev. (cm)	3.09	3.92	3.16	3.34

Table 3.4. Average minimum distance from obstacles over 50 randomly perturbed environments. Negative values indicate collisions.

Nonlinear Planar Quadrotor

For the nonlinear planar quadrotor system (2.33) from Chapter 1, the iterative P-GVIMP algorithm converges within 10-20 outer iterations. Figure 3.7 shows results for four different planning tasks with obstacles. The algorithm successfully plans collision-free trajectories while accounting for the nonlinear attitude dynamics.

The norm difference between consecutive linearizations decreases from 286.40 (iteration 1) to 2.96 (iteration 20), confirming convergence. By incorporating entropy regularization, the algorithm trades off trajectory length against robustness—for



(a) P-GVIMP (collision-free)

(b) GPMP2 (collision-free)

(c) BIT* (collision)

(d) FMT* (collision)

Figure 3.6. Comparison of planners under perception noise. Green obstacles show nominal environment, yellow ones show perturbed positions. P-GVIMP and GPMP2 remain collision-free, while sampling-based planners collide with disturbed obstacles.

instance, choosing to fly around rather than through narrow gaps when the entropy cost favors safety.

3.5 Discussion and Connections

The P-GVIMP framework bridges several key algorithmic paradigms:

- **Proximal methods:** The KL-proximal algorithm extends classical proximal point methods to distribution spaces, offering improved stability over natural

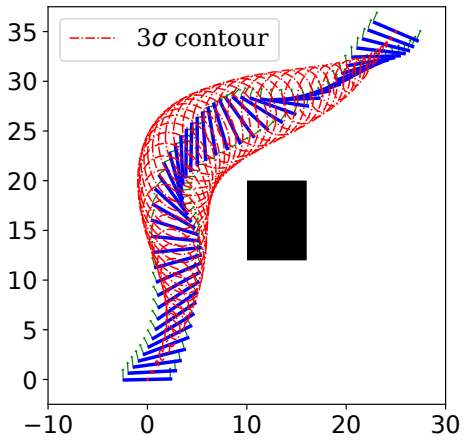


Figure 3.7. Iterative P-GVIMP result for nonlinear planar quadrotor. Red ellipses show 3σ position uncertainty along the trajectory.

gradient descent while exploiting the splitting structure of motion planning objectives.

- **Factor graphs:** By leveraging sparse factor graph structures and GBP, the algorithm scales efficiently to dense trajectory representations required for continuous-time planning.
- **Parallel computation:** The factorized gradient updates enable GPU-parallel collision checking, shifting the computational bottleneck from collision evaluation to marginal covariance computation.
- **Iterative linearization:** For nonlinear systems, the SLR-based outer loop parallels classical DDP/iLQG paradigms but operates directly on trajectory distributions rather than nominal trajectories.

Table 3.5 summarizes the key differences between the original GVIMP formulation (Chapter 1) and the proposed P-GVIMP framework.

Aspect	GVIMP	P-GVIMP
Algorithm	Natural gradient	KL-proximal
Complexity	$O(Nn^{k_q})$	$O(Nn^3)$
Hardware	CPU (serial)	GPU (parallel)
Nonlinear	Iterative LQG	Iterative SLR
Marginals	Direct evaluation	GBP

Table 3.5. Comparison between GVIMP and P-GVIMP frameworks.

3.6 Chapter Summary

This chapter developed an efficient computational framework for Gaussian variational inference motion planning by introducing a KL-proximal algorithm and exploiting parallel computation on GPUs. The key contributions are:

1. A KL-proximal point algorithm with splitting that provides stable, convergent updates for trajectory distribution optimization with explicit first-order approximation bounds.

2. Distributed gradient computation leveraging sparse factor graph structures and Gaussian Belief Propagation, reducing marginal covariance computation from $O((Nn)^3)$ to $O(Nn^3)$.

3. GPU-parallel collision checking that shifts the computational bottleneck from collision evaluation to covariance propagation, achieving over 96% speedup on high-dimensional systems.

4. An iterative framework for nonlinear systems combining Statistical Linear Regression with P-GVIMP as an efficient subroutine.

The experimental results demonstrate that P-GVIMP achieves near-real-time performance on 7-DOF robot arms with dense trajectories ($N = 750$), while maintaining robust performance under perception uncertainties. For nonlinear systems, the iterative SLR-based approach converges reliably within 10-20 iterations. These computational advances enable the practical deployment of trajectory distribution planning in time-critical robotic applications.

Chapter *4*

STOCHASTIC CONTROL FOR HYBRID SYSTEMS

4.1 Introduction

Many robotic systems exhibit hybrid dynamics comprising continuous-time flows and discrete-time transitions. Walking robots alternate between stance and flight phases (Collins and Ruina, 2005; Westervelt et al., 2018), running gaits involve complex contact sequences (Hutter et al., 2011), and manipulation tasks require making and breaking contact with objects (Johnson, Burden, and Koditschek, 2016; Billard and Kragic, 2019). These systems cannot be adequately modeled by purely continuous or discrete dynamics alone—they inherently combine both.

Uncertainty profoundly impacts hybrid systems through multiple channels. Modeling errors in contact dynamics, actuator noise, and external disturbances affect the continuous flows. Simultaneously, uncertainty in contact timing, impact configurations, and terrain properties introduces stochasticity into the discrete transitions. The interplay between these sources creates unique challenges: small perturbations before a discrete event can be amplified through the transition, potentially destabilizing limit cycles and degrading performance (Manchester, Tobenkin, et al., 2011; Manchester, Mettin, et al., 2011).

This chapter extends the stochastic optimal control framework developed in Chapters 2 and 3 to hybrid dynamical systems. We develop two complementary approaches unified by a path-distribution control perspective:

- **Hybrid Covariance Steering (H-CS):** For systems with linear stochastic flows, we formulate a convex optimization problem that steers both mean and covariance through hybrid transitions with prescribed boundary conditions. The approach leverages saltation matrices to propagate uncertainty through discrete jumps.
- **Hybrid Path Integral Control (H-PI):** For systems with nonlinear stochastic flows, we derive optimal controllers via forward sampling of trajectories subject to hybrid transitions, using importance sampling to reduce variance.

Both methods treat hybrid executions as path distributions over piecewise-smooth trajectories, enabling principled uncertainty quantification and control. The remainder of this chapter is organized as follows. Section 4.2 introduces hybrid dynamical systems and saltation matrix approximations. Section 4.3 develops the H-CS framework for linear flows. Section 4.5 presents the H-PI method for nonlinear flows. Section 4.4 provides experimental validation on bouncing ball and SLIP dynamics. Section 4.6 discusses the connections between the two approaches and their relative merits. Section 4.7 concludes.

4.2 Hybrid Dynamical Systems and Problem Formulation

Hybrid System Definition

A hybrid dynamical system is defined by a tuple (Grossman et al., 1993; Johnson, Burden, and Koditschek, 2016):

$$\mathcal{H} \coloneqq \{\mathcal{I}, \mathcal{D}, \mathcal{F}, \mathcal{G}, \mathcal{R}\}. \tag{4.1}$$

The set

$$\mathcal{I} \coloneqq \{I_1, I_2, \ldots, I_{N_I}\} \subset \mathbb{N} \tag{4.2}$$

is a finite set of *modes* representing discrete system configurations. $\mathcal{D}$ is the set of continuous *domains* containing the state spaces, with domain D_j for mode I_j. $\mathcal{F}$ is the set of *flows*, consisting of individual flow F_j that describes the smooth dynamics in mode I_j. $\mathcal{G} \subseteq \mathcal{D}$ denotes the set of *guards* triggering the resets, and $\mathcal{R}$ defines the *reset maps* between modes.

This formulation captures diverse robotic systems:

- **Legged Locomotion:** Modes represent contact configurations (e.g., single support, double support), guards detect foot contact/liftoff events, and resets model impact dynamics.
- **Manipulation:** Modes correspond to contact states (free motion, contact maintained, sliding), with guards detecting contact formation/breaking.
- **Multi-Robot Systems:** Modes represent coordination patterns, with discrete transitions between cooperative behaviors.

Stochastic Flows and Transitions

We denote $X_j(t) \in \mathbb{R}^{n_j}$ as the continuous-time state in mode I_j, and $u_j(t) \in \mathbb{R}^{m_j}$ as the state-dependent control input. In mode I_j, the flow F_j is considered to be a

linear time-varying stochastic process:

$$dX_j(t) = (A_j(t)X_j(t) + B_j(t)u_j(t))dt + \sqrt{\epsilon}B_j(t)dW_j(t), \tag{4.3}$$

where $A_j(t) \in \mathbb{R}^{n_j \times n_j}$ and $B_j(t) \in \mathbb{R}^{n_j \times m_j}$ are the system matrices in mode I_j, and $dW_j(t) \in \mathbb{R}^{m_j}$ denotes a standard Wiener process with noise intensity ϵ. All variables are *mode-dependent*, reflecting the structural changes in dynamics across discrete transitions.

A transition from mode I_j to mode I_k occurs at state $X_j(t^-)$ and time t^- when the guard condition is satisfied:

$$g_{jk}(t^-, X_j(t^-)) \leq 0, \quad X_j(t^-) \in I_j. \tag{4.4}$$

The guard function $g_{jk} : \mathbb{R} \times D_j \to \mathbb{R}$ defines a switching surface in the augmented state-time space. When the trajectory crosses this surface, an instantaneous jump occurs.

At the event time t^-, the reset map applies:

$$t^+ = t^-, \quad X_k(t^+) = R_{jk}(X_j(t^-)), \tag{4.5}$$

where $R_{jk} : D_j \to D_k$ is generally nonlinear. For legged robots, R_{jk} models impact dynamics; for manipulation, it represents instantaneous velocity changes upon contact.

Saltation Matrices and Uncertainty Propagation

The nonlinearity of reset maps R_{jk} complicates uncertainty propagation through discrete transitions. Saltation matrices (Kong, Payne, Zhu, et al., 2024) provide precise linear approximations for covariance propagation. The saltation matrix $\Xi_{jk} \in \mathbb{R}^{n_k \times n_j}$ from mode I_j to I_k is defined as:

$$\Xi_{jk} \triangleq \partial_x R_{jk} + \frac{(F_k - \partial_x R_{jk} \cdot F_j - \partial_t R_{jk})\partial_x g_{jk}}{\partial_t g_{jk} + \partial_x g_{jk} \cdot F_j}, \tag{4.6}$$

where $\partial_x(\cdot)$ and $\partial_t(\cdot)$ denote partial derivatives with respect to state and time.

The saltation matrix captures how perturbations in the pre-impact state map to post-impact perturbations:

$$\delta X(t^+) \approx \Xi_{jk}\delta X(t^-). \tag{4.7}$$

This first-order approximation has been successfully applied to controller design (Kong, Li, et al., 2023) and state estimation (Kong, Payne, Council, et al., 2021) for hybrid

systems. The approximation quality depends on the nonlinearity of the reset map and the magnitude of state perturbations.

For stochastic systems, we extend this approximation to covariance propagation. If $X_j(t^-) \sim \mathcal{N}(m^-, \Sigma^-)$, then to first order:

$$X_k(t^+) \sim \mathcal{N}(R_{jk}(m^-), \Xi_{jk}\Sigma^-\Xi'_{jk}). \tag{4.8}$$

This Gaussian approximation preserves tractability while accounting for uncertainty amplification (or attenuation) through impacts.

Assumptions and Notations

For clarity of exposition, we focus on systems with a single jump event, though the framework extends to multiple jumps. Consider the time window $[0, T]$ with one jump at $t = t^-$, separating the total time into two intervals. Denote the initial and terminal times in each interval as t_0^j and t_f^j respectively, with:

$$t_0^1 = 0, \quad t_f^1 = t^-, \quad t_0^2 = t^+, \quad t_f^2 = T. \tag{4.9}$$

The two modes before and after the jump event are I_1 and I_2, with pre-event states $X_1(t) \in \mathbb{R}^{n_1}$ for $t \leq t^-$ and post-event states $X_2(t) \in \mathbb{R}^{n_2}$ for $t \geq t^+$. We use

$$\Xi \triangleq \Xi_{12} \in \mathbb{R}^{n_2 \times n_1} \tag{4.10}$$

to represent the saltation transition from mode I_1 to I_2 at $t^- = t^+$.

This simplified setting captures the essential challenges of hybrid stochastic control while maintaining notational clarity. Extension to multiple jumps follows by iterative application of the single-jump results.

4.3 Hybrid Covariance Steering

Problem Formulation

The hybrid covariance steering problem seeks optimal control laws that steer both the mean trajectory and its associated covariance through a hybrid transition to prescribed terminal values. The formulation is:

$$\min_{u_j(t)} \mathcal{J}_H \triangleq \mathbb{E}\left\{\int_0^T \left[\|u_j(t)\|^2 + X_j'(t)Q_j(t)X_j(t)\right] dt\right\} \tag{4.11a}$$

$$\text{s.t.} \quad dX_1 = A_1(t)X_1 dt + B_1(t)(u_1 dt + \sqrt{\epsilon} dW_1), \tag{4.11b}$$

$$X_2(t^+) = \Xi X_1(t^-), \tag{4.11c}$$

$$dX_2 = A_2(t)X_2 dt + B_2(t)(u_2 dt + \sqrt{\epsilon} dW_2), \tag{4.11d}$$

$$X_1(0) \sim \mathcal{N}(m_0, \Sigma_0), \quad X_2(T) \sim \mathcal{N}(m_T, \Sigma_T). \tag{4.11e}$$

This formulation differs from standard LQG in two key aspects:

1. **Hybrid Dynamics:** The state space switches discontinuously at t^-, with dimensions potentially changing ($n_1 \neq n_2$ in general).

2. **Terminal Covariance Constraint:** In addition to mean constraints, the terminal covariance must satisfy $\mathrm{Cov}[X_2(T)] = \Sigma_T$.

The objective $\mathcal{J}_H$ penalizes control effort and state deviations, with mode-dependent weighting $Q_j(t)$ allowing different cost structures in each mode.

Path Measure Duality

Following the distributional control perspective, we reformulate H-CS in terms of path measures. Let $\mathbb{P}_j^*$ denote the measure induced by the unconstrained LQG problem in mode I_j:

$$\min_{\hat{u}_j(t)} \ \mathbb{E}\left\{ \int_{t_0^j}^{t_f^j} \left[\|\hat{u}_j(t)\|^2 + X_j(t)' Q_j(t) X_j(t) \right] dt \right\}. \tag{4.12}$$

The unconstrained optimal measure $\mathbb{P}_j^*$ serves as a reference distribution. The constrained problem then seeks the path measure $\mathbb{P}_j$ closest to $\mathbb{P}_j^*$ (in KL divergence) while satisfying boundary covariance constraints.

Denote the joint marginal distribution at initial and terminal times under path measure $\mathbb{P}_j$ by $\mathbb{P}_{(t_0^j, t_f^j)}$, and similarly for $\mathbb{P}_j^*$. As shown in (Yu, Franco, et al., 2024a), the covariance steering objective in mode I_j converts to:

$$\mathcal{J}_j = \mathrm{KL}\left(\mathbb{P}_{(t_0^j, t_f^j)} \parallel \mathbb{P}^*_{(t_0^j, t_f^j)} \right). \tag{4.13}$$

This finite-dimensional representation replaces integration over infinite-dimensional path space with optimization over marginal distributions—a dramatic simplification enabling tractable computation.

Covariance Propagation Through Jumps

The key challenge in hybrid covariance steering is propagating uncertainty through discrete transitions. Using the saltation approximation (4.7), the covariance constraint at the jump becomes:

$$\Sigma^+ = \Xi \Sigma^- \Xi', \tag{4.14}$$

where $\Sigma^- = \mathrm{Cov}[X_1(t^-)]$ and $\Sigma^+ = \mathrm{Cov}[X_2(t^+)]$.

This constraint couples the two modes: the pre-impact covariance Σ^- completely determines the post-impact covariance Σ^+ via the linear transformation Ξ. The coupling introduces dependencies absent in purely continuous systems, requiring careful treatment in the optimization.

Convex Reformulation via Semidefinite Programming

The H-CS problem admits a convex reformulation as a semidefinite program (SDP). Introduce decision variables W_1, W_2 (cross-covariances), Σ^-, Σ^+ (pre/post-jump covariances), and Y_1, Y_2 (slack variables). The convex formulation reads:

$$\min_{W_1,W_2,\Sigma^-,\Sigma^+,Y_1,Y_2} \mathcal{J}_{\mathrm{SDP}} \tag{4.15a}$$

$$\text{s.t.} \quad \Sigma^- \succ 0,\ Y_1 \succ 0,\ \Sigma^+ \succeq 0,\ Y_2 \succeq 0, \tag{4.15b}$$

$$\Sigma^+ = \Xi\Sigma^-\Xi', \tag{4.15c}$$

$$\begin{bmatrix} \Sigma^+ & W_2' \\ W_2 & \Sigma_T - Y_2 \end{bmatrix} \succeq 0, \tag{4.15d}$$

where the objective is (Yu, Franco, et al., 2024a):

$$\begin{aligned} \mathcal{J}_{\mathrm{SDP}} =& \frac{1}{\epsilon}\,\mathrm{Tr}(S_1^{-1}\Sigma^-) - \frac{2}{\epsilon}\,\mathrm{Tr}(\Phi'_{A_2}S_2^{-1}W_2) - \frac{2}{\epsilon}\,\mathrm{Tr}(\Phi'_{A_1}S_1^{-1}W_1) \\ &+ \frac{1}{\epsilon}\,\mathrm{Tr}(\Phi'_{A_2}S_2^{-1}\Phi_{A_2}\Sigma^+) - \log\det(Y_1) - \log\det(Y_2). \end{aligned} \tag{4.16}$$

Here Φ_{A_j} denotes the state transition matrix for mode I_j, and S_j are Gramians encoding process noise accumulation. The terms involving traces and log-determinants arise from the KL divergence structure, while the LMI constraints (4.15d) enforce realizability conditions.

Convexity and Computation. The formulation (4.15) is convex in its decision variables. The equality constraint (4.15c) is affine, and the objective combines convex trace terms with concave log-determinants (ensuring overall convexity through proper signs). Standard SDP solvers (e.g., CVXPY, MOSEK) efficiently find global optima.

Controller Recovery

Once the SDP is solved, the optimal covariances Σ^{-*}, Σ^{+*} and cross-covariances W_1^*, W_2^* are known. The feedback controller in each mode is recovered via Riccati-

like equations. In mode I_j, the optimal control law takes the form:

$$u_j(t) = -K_j(t)X_j(t), \tag{4.17}$$

where the gain $K_j(t)$ is computed by integrating hybrid Riccati equations backward from the optimal covariances.

The hybrid Riccati equations account for the jump at t^-:

$$\dot{P}_2(t) = -P_2A_2 - A_2'P_2 + P_2B_2B_2'P_2 - Q_2, \quad t \in (t^+, T], \tag{4.18a}$$

$$P_1(t^-) = \Xi' P_2(t^+)\Xi, \tag{4.18b}$$

$$\dot{P}_1(t) = -P_1A_1 - A_1'P_1 + P_1B_1B_1'P_1 - Q_1, \quad t \in [0, t^-), \tag{4.18c}$$

with terminal condition $P_2(T)$ determined from Σ^{+*} and Σ_T.

The saltation matrix Ξ appears in the jump condition, propagating cost-to-go information backward through the discrete transition. This coupling ensures the controller anticipates the impact dynamics when regulating pre-jump behavior.

4.4 Experiments

Experimental Validation: Bouncing Ball

We validate H-CS on a 1D bouncing ball with elastic impacts. The pre-impact dynamics are:

$$d\begin{bmatrix} p \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} p \\ v \end{bmatrix} dt + \begin{bmatrix} 0 \\ -g \end{bmatrix} dt + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (udt + \sqrt{\epsilon} dW), \tag{4.19}$$

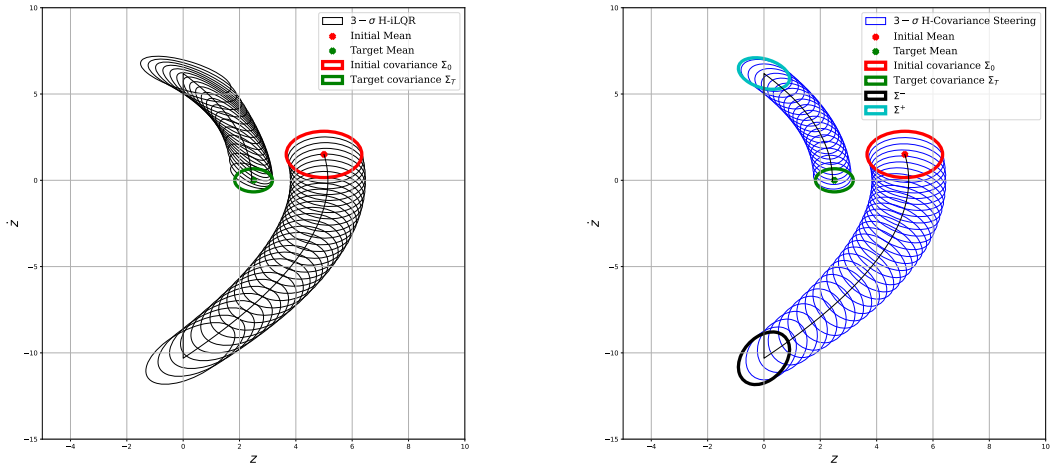
where p is height and v is velocity. The guard $g(p, v) = p$ triggers a bounce when $p \leq 0$, with reset:

$$\begin{bmatrix} p^+ \\ v^+ \end{bmatrix} = \begin{bmatrix} 0 \\ -ev^- \end{bmatrix}, \tag{4.20}$$

where $e \in (0, 1)$ is the coefficient of restitution. The saltation matrix is:

$$\Xi = \begin{bmatrix} 0 & 0 \\ 0 & -e \end{bmatrix}. \tag{4.21}$$

Figure 4.1 shows results comparing H-iLQR (standard hybrid LQR without covariance control) and H-CS. The H-iLQR controller (Figure 4.1(a)) produces a nominal trajectory with associated covariance, but this covariance is not controlled—it evolves passively under the linearized dynamics. In contrast, H-CS



(a) H-iLQR (passive covariance)

(b) H-CS (controlled covariance)

Figure 4.1. Covariance steering for bouncing ball dynamics with elastic impacts. H-CS guarantees the terminal covariance constraint, while H-iLQR's covariance drifts from the target (black vs. colored trajectories).

(Figure 4.1(b)) actively steers the covariance to match the prescribed terminal distribution $\mathcal{N}(m_T, \Sigma_T)$.

The key observation: H-CS precisely achieves $\mathrm{Cov}[X_2(T)] = \Sigma_T$, whereas H-iLQR exhibits terminal covariance mismatch. This control over uncertainty is critical for tasks requiring probabilistic guarantees (e.g., ensuring 95% of trajectories remain within safety bounds).

Experimental Validation: SLIP Model

The Spring-Loaded Inverted Pendulum (SLIP) model provides a more challenging test case with nonlinear reset maps. SLIP consists of a point mass m connected to a massless spring leg of rest length l_0 and stiffness k. The state is $(x, y, \dot{x}, \dot{y})$ in Cartesian coordinates.

During the stance phase (foot on ground), the dynamics are:

$$\begin{aligned} m\ddot{x} &= -k(l - l_0)\frac{x - x_f}{l} + u_x, \\ m\ddot{y} &= -k(l - l_0)\frac{y}{l} - mg + u_y, \end{aligned} \tag{4.22}$$

where $l = \sqrt{(x - x_f)^2 + y^2}$ is leg length and $(x_f, 0)$ is the foot position. The guard $g = y - l_0$ triggers liftoff when the leg reaches rest length.

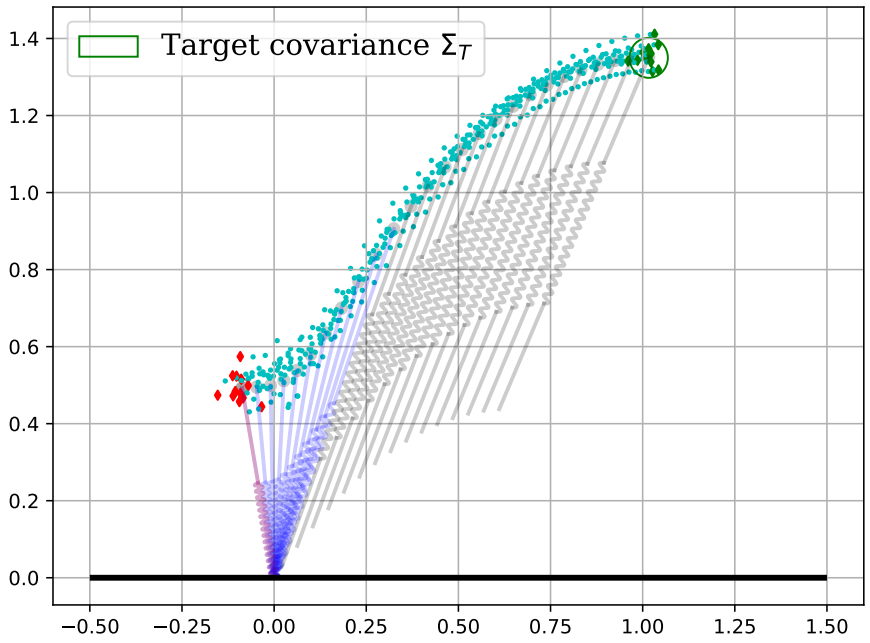


Figure 4.2. Deterministic nominal trajectory under H-iLQR controller (black) and stochastic sample trajectories under the H-CS controller (colored) for the SLIP model. The covariance tube narrows and expands according to the prescribed covariance profile.

During flight, the dynamics reduce to ballistic motion:

$$\ddot{x} = 0, \quad \ddot{y} = -g. \tag{4.23}$$

Landing occurs when $y \leq 0$ with leg angle satisfying a prescribed touchdown condition.

We apply H-CS by linearizing around a nominal SLIP trajectory obtained via H-iLQR. Figure 4.2 shows sample trajectories under the H-CS controller, demonstrating successful covariance control through multiple stance-flight transitions.

The samples exhibit tight clustering around the nominal initially, spread during ballistic flight (where control authority is limited), and reconverge after landing—a behavior directly controlled by the covariance prescription Σ_T. This demonstrates H-CS's applicability beyond simple academic examples to realistic locomotion models.

4.5 Hybrid Path Integral Control

Motivation and Overview

While H-CS provides elegant convex solutions for linear flows, many robotic systems exhibit nonlinear stochastic dynamics where linearization may be inadequate.

Path integral control (Kappen, Gómez, and Opper, 2012) offers a sampling-based alternative that directly handles nonlinearity through forward trajectory rollouts.

This section extends path integral control to hybrid systems, deriving optimal controllers for nonlinear flows subject to discrete transitions. The key insight: the Girsanov change-of-measure formula extends naturally to hybrid path distributions, enabling importance sampling-based optimization.

Nonlinear Hybrid Dynamics

For the hybrid system (4.2) with guards (4.4) and resets (4.5), consider mode-dependent state $X_t^j \in \mathbb{R}^{n_j}$ and control $u_t^j \in \mathbb{R}^{m_j}$ variables in mode I_j. The *controlled* smooth flow is:

$$dX_t^j = F_j(t, X_t^j)dt + \sigma_j(t, X_t^j)(u^j(t, X_t^j)dt + \sqrt{\epsilon}dW_t^j), \tag{4.24}$$

where F_j is the drift and σ_j is the diffusion coefficient (potentially state-dependent).

The *uncontrolled* smooth flow in mode I_j is:

$$dX_t^j = F_j(t, X_t^j)dt + \sqrt{\epsilon}\sigma_j(t, X_t^j)dW_t^j. \tag{4.25}$$

A trajectory realization $\{X_t | t \in [0, T]\}$ experiences N_J hybrid transitions at times $\{t_j^-\}_{j=1,\dots,N_J}$. We extend the jump time set by defining $t_0^+ = 0$ and $t_{N_J+1}^- = T$. The system occupies mode I_j during $t \in [t_j^+, t_{j+1}^-]$:

$$X_t^j \in I_j, \ \forall t \in [t_j^+, t_{j+1}^-]; \quad X^{j+1}(t_{j+1}^+) = R_{j,j+1}(X^j(t_j^-)). \tag{4.26}$$

Hybrid Stochastic Optimal Control Problem

The control problem over the time window $[0, T] = \bigcup_{j=0}^{N_J}[t_j^+, t_{j+1}^-]$ is:

$$\min_u \mathcal{J}_H \triangleq \mathbb{E}\left[\sum_{j=0}^{N_J} \int_{t_j^+}^{t_{j+1}^-} \left(V(t, X_t^j) + \frac{1}{2}\|u_t^j\|^2\right) dt + \Psi_T\right] \tag{4.27}$$
$$\text{s.t. (4.24), (4.4), (4.5), } \forall j = 1, \dots, N_I.$$

The objective accumulates running cost $V(t, X_t^j)$ (e.g., deviation from desired trajectory) and control effort $\frac{1}{2}\|u_t^j\|^2$ across all modes, plus a terminal cost Ψ_T.

This formulation is significantly more general than H-CS:

- Nonlinear drift F_j and diffusion σ_j

- State-dependent control authority via σ_j
- Arbitrary running costs V (not necessarily quadratic)
- Multiple jumps ($N_J > 1$)

The price for this generality is loss of convexity—no closed-form solution exists. Path integral control provides a sampling-based approximation.

Girsanov Theorem for Hybrid Processes

The key technical tool is the Girsanov change-of-measure formula, which relates controlled and uncontrolled path distributions. Let $\mathbb{P}^u$ denote the measure induced by controlled process (4.24) and $\mathbb{P}^0$ the measure induced by uncontrolled process (4.25), both with identical jump dynamics (4.5).

The Radon-Nikodym derivative is (Yu, Franco, et al., 2024b):

$$\frac{d\mathbb{P}^u}{d\mathbb{P}^0} = \exp\left(\sum_{j=0}^{N_J} \int_{t_j^+}^{t_{j+1}^-} -\frac{\|u_t^j\|^2}{2\epsilon} dt + \frac{1}{\sqrt{\epsilon}} (u_t^j)' dW_t^j \right) \tag{4.28a}$$

$$= \exp\left(\sum_{j=0}^{N_J} \int_{t_j^+}^{t_{j+1}^-} \frac{1}{2\epsilon} \|u_t^j\|^2 dt + \frac{1}{\sqrt{\epsilon}} (u_t^j)' d\tilde{W}_t^j \right) \tag{4.28b}$$

$$=: \exp\left(\frac{1}{\epsilon} \Lambda_H\right), \tag{4.28c}$$

where dW_t is a Wiener process under $\mathbb{P}^0$, and $\tilde{W}_t$ is a Wiener process under $\mathbb{P}^u$.

Key Observation. The hybrid jump dynamics *do not appear* in the Radon-Nikodym derivative. This follows because the jumps are deterministic conditional on the pre-jump state. Both $\mathbb{P}^u$ and $\mathbb{P}^0$ apply identical resets, so the likelihood ratio depends only on the smooth flow between jumps.

This insight enables path integral control for hybrid systems: sampling can proceed as in purely continuous systems, simply accounting for deterministic jumps when they occur.

Cross-Entropy Formulation

Following (Kappen, Gómez, and Opper, 2012; Zhang et al., 2014), we reformulate control synthesis as distribution matching. Let $\mathbb{P}^*$ denote the optimal controlled

measure solving (4.27). The optimal controller minimizes the KL divergence:

$$\begin{aligned} u^* &= \arg\min_u \mathrm{KL}\left(\mathbb{P}^* \parallel \mathbb{P}^u\right) \\ &= \arg\min_u \mathbb{E}_{\mathbb{P}^*}\left[\log \frac{d\mathbb{P}^*}{d\mathbb{P}^0}\frac{d\mathbb{P}^0}{d\mathbb{P}^u}\right] \\ &= \arg\min_u \mathbb{E}_{\mathbb{P}^*}\left[-\frac{1}{\epsilon}\Lambda_H\right], \end{aligned} \tag{4.29}$$

since the term $\log \frac{d\mathbb{P}^*}{d\mathbb{P}^0}$ is independent of the control u.

This transformation converts the original stochastic optimal control problem into a distribution matching problem in path space—a perspective that enables sampling-based approximation.

Path Integral Optimal Controller

The optimal controller for (4.29) at time t can be expressed as an expectation over future trajectories:

$$u_t^* = \frac{\sqrt{\epsilon} \times \mathbb{E}_{\mathbb{P}^0}\left[\Delta W_t \exp\left(-\frac{1}{\epsilon}\mathcal{L}_H(t)\right)\right]}{\Delta t \times \mathbb{E}_{\mathbb{P}^0}\left[\exp\left(-\frac{1}{\epsilon}\mathcal{L}_H(t)\right)\right]}, \tag{4.30}$$

where $\mathcal{L}_H(t)$ denotes the cost-to-go from time t onward:

$$\mathcal{L}_H(t) = \sum_{j:t_j^+ \geq t} \int_{t_j^+}^{t_{j+1}^-} V(s, X_s^j)ds + \Psi_T. \tag{4.31}$$

Equation (4.30) provides the theoretical optimal control, but requires sampling from the (unknown) uncontrolled measure $\mathbb{P}^0$ and evaluating cost-to-go for exponentially many future hybrid transitions—computationally intractable.

Importance Sampling and Variance Reduction

To make (4.30) tractable, we employ importance sampling. Replace the uncontrolled measure $\mathbb{P}^0$ with a proposal measure $\mathbb{P}^u$ induced by a nominal controller (e.g., H-iLQR):

$$\begin{aligned} \mathbb{E}_{\mathbb{P}^0}\left[\Delta W_t \exp\left(-\frac{\mathcal{L}_H}{\epsilon}\right)\right] &= \mathbb{E}_{\mathbb{P}^u}\left[\Delta W_t \frac{d\mathbb{P}^0}{d\mathbb{P}^u}\exp\left(-\frac{\mathcal{L}_H}{\epsilon}\right)\right] \\ &= \mathbb{E}_{\mathbb{P}^u}\left[\Delta W_t \exp\left(-\frac{\mathcal{S}_H}{\epsilon}\right)\right], \end{aligned} \tag{4.32}$$

where $\mathcal{S}_H^u(t)$ is the importance-weighted cost-to-go:

$$\mathcal{S}_H^u(t) \triangleq \mathcal{I}_S[t, t_{j_m-1}^-] + \sum_{j=j_m}^{N_J} \mathcal{I}_S[t_j^+, t_{j+1}^-], \tag{4.33}$$

$$\mathcal{I}_S[t_j^+, t_{j+1}^-] \triangleq \int_{t_j^+}^{t_{j+1}^-} \left(\frac{1}{2}\|u_s^j\|^2 + V(s, X_s^j)\right) ds + \sqrt{\epsilon} \int_{t_j^+}^{t_{j+1}^-} (u_s^j)' d\tilde{W}_s^j + \Psi_T,$$

where $j_m = \min\{j : t_j^+ \geq t\}$.

The optimal control (4.30) becomes:

$$u_t^* = \frac{\sqrt{\epsilon} \times \mathbb{E}_{\mathbb{P}^u}\left[\Delta W_t \exp\left(-\frac{1}{\epsilon}\mathcal{S}_H(t)\right)\right]}{\Delta t \times \mathbb{E}_{\mathbb{P}^u}\left[\exp\left(-\frac{1}{\epsilon}\mathcal{S}_H(t)\right)\right]}. \tag{4.34}$$

Practical Implementation. To evaluate (4.34), we:

1. Sample K trajectories from $\mathbb{P}^u$ by forward rollout under nominal control with process noise
2. For each sample k, compute the weighted cost $\exp(-\mathcal{S}_H^{(k)}/\epsilon)$
3. Approximate the expectations via empirical averages:

$$u_t^* \approx \frac{\sqrt{\epsilon}}{\Delta t} \frac{\sum_{k=1}^K \Delta W_t^{(k)} \exp(-\mathcal{S}_H^{(k)}/\epsilon)}{\sum_{k=1}^K \exp(-\mathcal{S}_H^{(k)}/\epsilon)} \tag{4.35}$$

The quality of the proposal distribution $\mathbb{P}^u$ critically affects variance. A good proposal (H-iLQR provides one) concentrates samples in high-probability regions under the optimal measure, reducing the number of samples K required for accurate approximation.

Experimental Validation: Bouncing Ball

We apply H-PI to the nonlinear bouncing ball with state-dependent diffusion:

$$d\begin{bmatrix} p \\ v \end{bmatrix} = \begin{bmatrix} v \\ -g \end{bmatrix} dt + \begin{bmatrix} 0 \\ \sqrt{1+v^2} \end{bmatrix} (udt + \sqrt{\epsilon} dW), \tag{4.36}$$

where diffusion intensity increases with velocity. This models increasing actuator uncertainty during high-speed motion.

Table 4.1. Expected cost improvement over H-iLQR baseline (100 trials).

		H-iLQR	H-PI	Improved (%)
Expected Cost	Bouncing Ball	60.11	57.86	3.74
	SLIP	0.2459	0.2164	**12.00**
H-iLQR 10% tail	Bouncing Ball	141.65	115.40	**11.46**
	SLIP	0.9451	0.2616	**63.29**

Table 4.2. Conditional Value at Risk (CVaR) cost improvement at different confidence levels.

	Confidence level	0.7	0.8	0.9
Improvement (%)	Bouncing Ball	15.05	21.34	32.44
	SLIP	37.86	47.48	59.32

Table 4.1 shows results averaged over 100 random trials. H-PI achieves 3.74% expected cost reduction compared to H-iLQR. More significantly, in high-cost scenarios (worst 10% of H-iLQR trials), H-PI improves by 11.46%—demonstrating robustness in challenging situations.

The "tail" results (worst 10% of H-iLQR) are particularly informative. These represent scenarios where linearization-based control performs poorly—high nonlinearity, large disturbances, or unfavorable noise realizations. H-PI's sampling-based approach naturally handles these cases, achieving a dramatic 63.29% improvement for SLIP.

Experimental Validation: SLIP Dynamics

The SLIP model provides a challenging test with state-dependent nonlinearities and multiple stance-flight transitions. Table 4.1 shows H-PI achieves 12% average improvement and 63.29% improvement in the worst 10% of cases.

To understand this large tail improvement, we analyze the risk sensitivity via Conditional Value-at-Risk (CVaR). Table 4.2 shows cost improvements conditioned on confidence levels 70%, 80%, and 90%. For SLIP, improvements range from 37.86% (70% confidence) to 59.32% (90% confidence)—H-PI provides increasingly large benefits in high-risk scenarios.

This risk-sensitive improvement stems from H-PI's ability to handle nonlinearity without approximation. When H-iLQR's linearization becomes inaccurate (high-risk scenarios), H-PI's forward sampling remains valid.

Table 4.3. Average weight variance and effective sample size before/after jump events.

		$[0, T]$	$[0, t^-]$	$[t^-, T]$	**Jump Influence**
Bouncing Ball	Avg. Var(α)	0.64	0.78	0.25	67.95%
	Avg. λ(%)	69.59	65.83	79.85	21.2%
SLIP	Avg. Var(α)	14.20	34.03	3.55	89.57%
	Avg. λ(%)	64.69	10.32	93.89	809.8%

Importance Sampling Efficiency Analysis

The effectiveness of importance sampling depends on variance of the importance weights. We analyze two key metrics:

Weight Variance. Define the normalized importance weights:

$$\alpha^{(k)} = \frac{\exp(-\mathcal{S}_H^{(k)}/\epsilon)}{\sum_{i=1}^{K} \exp(-\mathcal{S}_H^{(i)}/\epsilon)}. \tag{4.37}$$

High variance in $\{\alpha^{(k)}\}$ indicates poor proposal quality—a few samples dominate, leading to high Monte Carlo error.

Effective Sample Size. The effective sample size is:

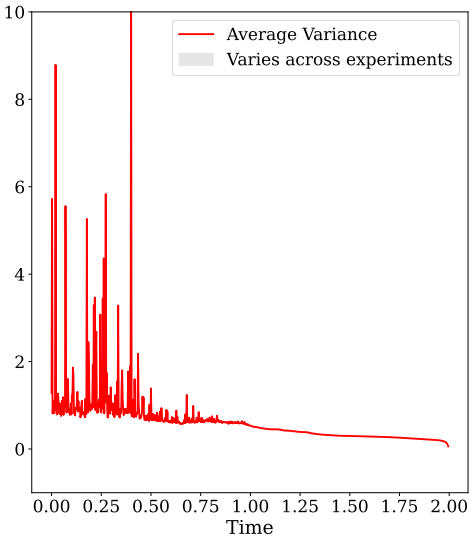
$$\lambda = \frac{1}{K \sum_{k=1}^{K} (\alpha^{(k)})^2} \in (0, 1]. \tag{4.38}$$

Values near 1 indicate efficient sampling; values near 0 indicate degeneracy (one sample dominates).

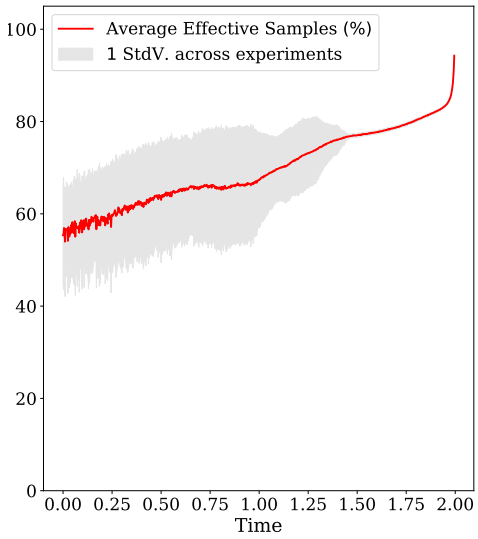
Figure 4.3 shows Var(α) and λ over time for both bouncing ball and SLIP. Key observations:

- **Before Jump:** High variance, low λ—the hybrid transition introduces uncertainty that the linearized proposal struggles to capture.
- **After Jump:** Lower variance, higher λ—post-impact dynamics better matched by the proposal.

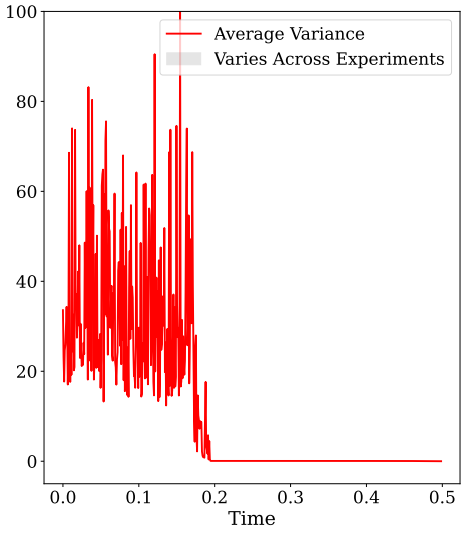
Table 4.3 quantifies this effect. For SLIP, λ increases from 10.32% pre-jump to 93.89% post-jump—an 809.8% relative increase. This dramatic change indicates that hybrid transitions pose significant challenges for linearization-based methods, motivating the sampling-based H-PI approach.



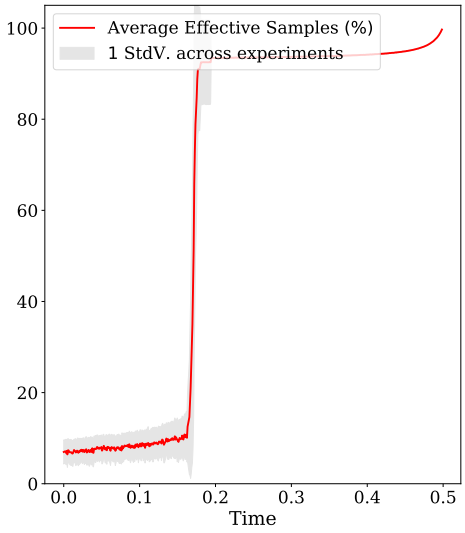
(a) Variance, bouncing ball



(b) Effective samples, ball



(c) Variance, SLIP



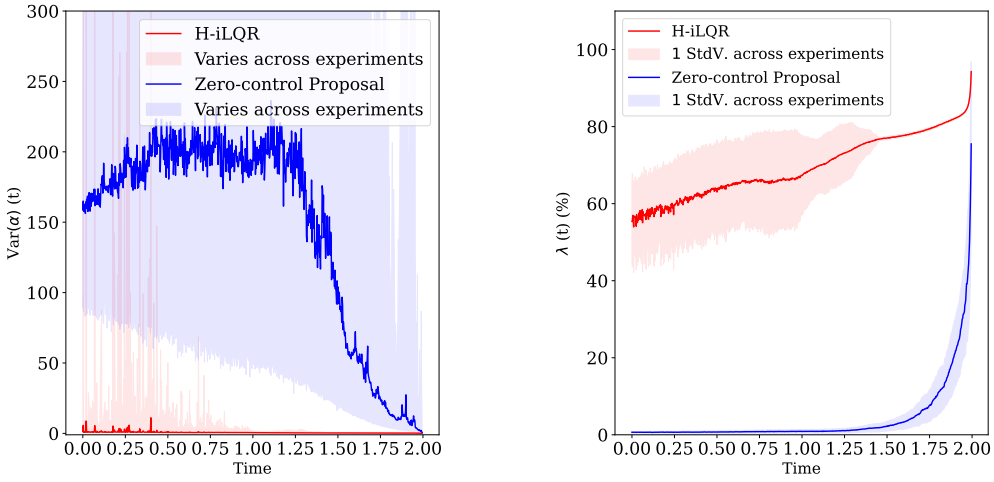
(d) Effective samples, SLIP

Figure 4.3. Sample variance and effective sample portions for bouncing ball and SLIP. Both indicators change dramatically before and after jump events, revealing the impact of hybrid transitions on controller robustness.

Comparison with Zero Control Proposal

To validate the importance of a good proposal distribution, we compare H-iLQR proposal versus zero control ($u = 0$) proposal. Figure 4.4 shows variance and effective sample size for both.

The zero control proposal exhibits orders of magnitude higher variance and near-zero effective sample size, rendering importance sampling impractical. This validates the



(a) Variance

(b) Effective Samples

Figure 4.4. Comparison of variance and effective samples for H-iLQR proposal (blue) versus zero control (orange). H-iLQR provides dramatically lower variance, enabling efficient importance sampling.

design choice: H-iLQR provides a high-quality proposal that concentrates samples near optimal trajectories, dramatically reducing Monte Carlo variance.

4.6 Discussion and Connections

Unified Path-Distribution Perspective

Both methods share a common conceptual foundation: viewing hybrid executions as path distributions and seeking optimal distributions under different criteria.

H-CS: Minimizes KL divergence to unconstrained LQG reference while satisfying terminal covariance constraints:

$$\min_{\mathcal{P}} \ \mathrm{KL}(\mathcal{P}\|\mathcal{P}^*) \quad \text{s.t.} \quad \mathrm{Cov}_{t=T}[\mathcal{P}] = \Sigma_T, \ \Sigma^+ = \Xi\Sigma^-\Xi'. \tag{4.39}$$

H-PI: Minimizes expected cost (equivalently, KL divergence to Boltzmann-weighted uncontrolled measure):

$$\min_{\mathcal{P}} \ \mathbb{E}_{\mathcal{P}}[\mathrm{Cost}] \equiv \min_{\mathcal{P}} \ \mathrm{KL}(\mathcal{P}\|\mathcal{P}_{\mathrm{Boltzmann}}). \tag{4.40}$$

This unification suggests potential hybrid approaches: use H-CS for coarse trajectory optimization with covariance control, then refine via H-PI sampling to account for unmodeled nonlinearities.

Connections to Literature

Hybrid Systems Control. Classical hybrid control (Grossman et al., 1993; Lygeros et al., 2003) focuses on deterministic systems. Our stochastic extension connects to recent work on stochastic hybrid systems (Tassa, 2011; Drnach and Y. Zhao, 2021) but provides explicit covariance control (H-CS) and sampling-based nonlinear control (H-PI).

Covariance Steering. Covariance steering for continuous systems (Chen., T. Georgiou, and M. Pavon, 2016; Y. Chen, Tryphon T Georgiou, and Michele Pavon, 2018) has been well-studied. Our saltation-based extension enables application to hybrid systems, bridging optimal transport theory (Caluya and Halder, 2020) with hybrid dynamics.

Path Integral Control. Path integral methods (Kappen, Gómez, and Opper, 2012; Theodorou, Buchli, and Schaal, 2010) typically assume purely continuous dynamics. Our Girsanov formula for hybrid processes enables direct extension to systems with discrete transitions, maintaining computational tractability.

Limitations and Future Directions

Deterministic Guards. Current formulations assume deterministic guard functions. In reality, contact timing involves uncertainty due to terrain irregularities, sensor noise, and model errors. Future work should incorporate stochastic guards, potentially via:

- Probabilistic guard functions $P(g_{jk}(t, X) \leq 0)$
- Robust optimization ensuring feasibility across guard uncertainty sets
- Learning guard distributions from data

Multiple Transitions. While the single-jump case provides clarity, real locomotion involves many contacts. Extending H-CS to multiple jumps requires iterative SDP solving or augmented state spaces. H-PI naturally handles multiple jumps but may require more samples for long horizons.

Learning and Adaptation. Both methods assume known models (linearized for H-CS, sampled for H-PI). Integrating learning—either of dynamics models, saltation matrices, or cost functions—would enhance applicability to unknown environments.

4.7 Chapter Summary

This chapter extended the stochastic optimal control framework to hybrid dynamical systems combining continuous flows with discrete transitions. Two complementary approaches were developed:

Hybrid Covariance Steering (H-CS). For linear stochastic flows, we formulated a convex SDP that steers mean and covariance through hybrid transitions with saltation matrix-based uncertainty propagation. The method provides global optimality and worst-case covariance guarantees. Experiments on bouncing ball and SLIP dynamics validated precise covariance control.

Hybrid Path Integral Control (H-PI). For nonlinear flows, we derived optimal controllers via forward sampling under a Girsanov change-of-measure. Importance sampling with H-iLQR proposals reduces variance. Experiments demonstrated 12% expected cost improvement and 63% improvement in high-risk scenarios, with risk-sensitive benefits quantified via CVaR analysis.

Key insights include:

- **Saltation Matrices:** Enable tractable uncertainty propagation through deterministic discrete transitions via linear approximation.
- **Path-Distribution Unification:** Both methods optimize over hybrid path distributions under different optimality criteria, suggesting potential hybrid approaches.
- **Importance Sampling Efficiency:** Hybrid transitions dramatically affect importance weight variance, motivating careful proposal distribution design.
- **Risk Sensitivity:** Sampling-based H-PI provides increasing benefits in high-risk scenarios where linearization breaks down.

Together with GVIMP (Chapter 2) and P-GVIMP (Chapter 3), these methods provide a comprehensive toolkit for probabilistic motion planning and control under uncertainty, spanning continuous smooth flows, nonlinear dynamics, GPU-accelerated inference, and hybrid discrete-continuous systems.

Chapter 5

CONCLUSION AND FUTURE WORK

5.1 Summary of Contributions

This dissertation developed a unified probabilistic framework for motion planning and control under uncertainty in robotic systems. The work addressed two fundamental challenges: (i) the computational intractability of solving high-dimensional trajectory distribution optimization problems, and (ii) the need for algorithms that achieve both performance optimality and robustness to uncertainty.

Gaussian Variational Inference Motion Planning

Chapter 2 introduced GVIMP, which reformulates motion planning as probabilistic inference over trajectory distributions. Key contributions include:

- **Planning-as-Inference Duality:** Establishing the connection between stochastic optimal control and probabilistic inference, enabling variational methods for trajectory optimization.
- **Natural Gradient Algorithm:** Developing a natural gradient descent that accounts for the Riemannian geometry of distribution space.
- **Sparse Factor Graph Exploitation:** Leveraging sparsity to decompose gradient computations into independent marginal expectations.
- **Closed-Form Prior Updates:** Deriving analytic gradient expressions for linear-Gaussian dynamics, avoiding expensive numerical quadrature.
- **Entropy Regularization:** Incorporating entropy maximization to promote robustness, connecting to maximum-entropy planning.

Section 2.6 developed a proximal gradient perspective in distribution space, connecting GVIMP to optimal transport and covariance steering. For nonlinear systems, an iterative linearization framework provides closed-form solutions at each step with provably linear convergence.

Hardware experiments on a Franka arm demonstrated superior robustness: GVIMP maintained collision-free execution across 50 random perturbations (average minimum distance: 5.46 cm), while baseline methods exhibited frequent collisions (negative distances).

Parallel Gaussian Variational Inference Motion Planning

Chapter 3 addressed GVIMP's computational bottleneck through algorithmic and architectural innovations:

- **KL-Proximal Algorithm:** Introducing a proximal point algorithm in distribution space with convergence guarantees, exploiting the splitting structure of motion planning objectives.
- **Gaussian Belief Propagation:** Leveraging GBP for efficient marginal covariance computation, reducing complexity from $O((Nn)^3)$ to $O(Nn^3)$ with over 99% speedup.
- **GPU Parallelization:** Implementing parallel collision checking achieving 96% speedup for 7-DOF systems with dense trajectories ($N = 750$).
- **Statistical Linear Regression Extension:** Developing an iterative framework for nonlinear systems alternating between SLR-based linearization and P-GVIMP subroutines.

Overall, P-GVIMP achieved 97% end-to-end speedup compared to serial GVIMP, enabling near-real-time performance on high-dimensional planning problems.

Stochastic Control for Hybrid Systems

Chapter 4 extended the framework to hybrid dynamical systems:

- **Hybrid Covariance Steering (H-CS):** For linear stochastic flows with saltation transitions, H-CS formulates a convex semidefinite program steering mean and covariance with guaranteed performance.
- **Hybrid Path Integral Control (H-PI):** For nonlinear flows, H-PI derives optimal control via forward sampling with importance sampling, showing 12% expected cost improvement and 63% improvement in high-cost scenarios.

5.2 Key Theoretical Insights

Duality Between Control and Inference

A central insight is the duality between stochastic optimal control and probabilistic inference. For linear-Gaussian systems, the optimal trajectory distribution can be characterized as a posterior combining prior dynamics with cost-augmented likelihoods. This enables:

- Distributional solutions that quantify uncertainty
- Variational approximations for tractable optimization
- Entropy regularization as robustness promotion

Sparse Structure and Computational Efficiency

The sparse factor graph structure—where dynamics couple only adjacent time steps and costs factorize across states—enables efficiency through:

- Marginal factorization decomposing global gradients into local expectations
- Message passing via GBP for efficient inference
- Parallelization distributing factor evaluations across GPU cores

Proximal Methods in Distribution Space

Extending proximal algorithms from Euclidean spaces to distribution spaces provides:

- Geometric awareness via KL divergence as proximal term
- Explicit updates exploiting objective splitting structure
- Provable linear convergence under standard assumptions

5.3 Limitations and Open Questions

Despite the contributions, several limitations merit acknowledgment:

Gaussian Approximation. Restricting to Gaussian distributions limits expressiveness:

- Multi-modal posteriors from multiple homotopy classes cannot be captured by single Gaussians
- Non-Gaussian uncertainty sources (contact dynamics, adversarial disturbances) may not be well-approximated

5.4 Future Research Directions

GPU-Native Implementation

P-GVIMP currently parallelizes collision checking and gradient computation across trajectory waypoints on GPU, similar to GPU-based iLQR and DDP (Lee, Cho, and Kim, 2022; Plancher and Kuindersma, 2018). However, the KL-proximal update remains on CPU, requiring expensive data transfers between CPU and GPU that limit overall performance. Implementing the entire optimization on GPU would eliminate this transfer bottleneck.

Diffusion-Based Variational Motion Planning

Recent advances in diffusion models (Ho, Jain, and Abbeel, 2020; Song et al., 2021) offer a promising direction for extending the variational motion planning framework beyond Gaussian approximations. Diffusion models learn to generate samples from complex, multi-modal distributions through iterative denoising, providing a natural path to capture the multi-modality inherent in motion planning problems.

Diffusion Process for Trajectory Distributions. A diffusion-based approach would model trajectory distributions through a forward noising process and reverse denoising process:

- **Forward Process:** Define a Markov chain gradually adding Gaussian noise to trajectories: $q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}) = \mathcal{N}(\mathbf{x}^{(t)}; \sqrt{1-\beta_t}\mathbf{x}^{(t-1)}, \beta_t\mathbf{I})$
- **Reverse Process:** Learn a denoising network $p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})$ that generates collision-free trajectories satisfying task objectives
- **Score Matching:** Train the model using score matching objectives that connect naturally to the GVIMP variational framework:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{t,\mathbf{x}^{(0)},\epsilon}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}^{(t)}, t)\|^2\right] \tag{5.1}$$

Multi-Modal Trajectory Generation. Unlike single Gaussians, diffusion models naturally capture multi-modal posteriors:

- **Homotopy Classes:** Different denoising paths naturally correspond to distinct homotopy classes around obstacles
- **Mode Discovery:** No need for explicit mode enumeration; modes emerge from the learned denoising process
- **Smooth Interpolation:** The diffusion process provides smooth interpolation between modes

This direction promises to combine GVIMP's principled probabilistic foundation with the flexibility and expressiveness of modern generative models, enabling robust planning in complex, multi-modal environments while maintaining computational tractability through learned representations.

5.5 Concluding Remarks

This dissertation demonstrates that motion planning under uncertainty can be formulated, approximated, and solved efficiently through probabilistic inference and distributional control. By connecting stochastic optimal control to variational inference, exploiting sparse structures, and leveraging parallel computing, the developed methods achieve both theoretical rigor and practical efficiency.

Several themes emerge:

Theory and Practice. The progression from GVIMP's natural gradient to P-GVIMP's GPU implementation shows that principled probabilistic methods can achieve computational feasibility through careful exploitation of problem structure.

Complementary Perspectives. Viewing problems through multiple lenses—variational inference, distributional control, optimal transport—provides algorithmic alternatives and deeper understanding. Different formulations excel in different regimes.

Uncertainty as Resource. Rather than minimizing uncertainty, entropy regularization reveals it as promoting robustness. Distributional solutions embracing uncertainty often outperform deterministic solutions ignoring it.

Structure and Efficiency. The sparse factor graph structure reflects fundamental locality in planning problems. Methods respecting this structure—message passing, marginal factorization, parallel evaluation—achieve dramatic efficiency gains.

From Smooth to Hybrid. The path-distribution perspective provides a unifying framework treating smooth and discrete components consistently, with practical importance for legged locomotion and manipulation.

Looking forward, integrating probabilistic planning with learning promises to enhance autonomous capabilities. As robots operate in increasingly complex and uncertain environments, explicit uncertainty reasoning becomes essential. The methods and insights developed here provide a foundation for truly robust, safe, and capable autonomous systems under pervasive uncertainty.

BIBLIOGRAPHY

Amari, Shun-Ichi (1998). "Natural gradient works efficiently in learning". In: *Neural computation* 10.2, pp. 251–276.

Arasaratnam, Ienkaran, Simon Haykin, and Robert J Elliott (2007). "Discrete-time nonlinear filtering algorithms using Gauss-Hermite quadrature". In: *Proceedings of the IEEE* 95.5, pp. 953–977.

Åström, Karl J (2012). *Introduction to stochastic control theory*. Courier Corporation.

Barfoot, Tim D, Chi Hay Tong, and Simo Särkkä (2014). "Batch Continuous-Time Trajectory Estimation as Exactly Sparse Gaussian Process Regression." In: *Robotics: Science and Systems*. Vol. 10. Citeseer, pp. 1–10.

Barfoot, Timothy D (2024). *State estimation for robotics*. Cambridge University Press.

Bertsekas, Dimitri and Steven E Shreve (1996). *Stochastic optimal control: the discrete-time case*. Vol. 5. Athena Scientific.

Billard, Aude and Danica Kragic (2019). "Trends and challenges in robot manipulation". In: *Science* 364.6446, eaat8414.

Botvinick, Matthew and Marc Toussaint (2012). "Planning as inference". In: *Trends in cognitive sciences* 16.10, pp. 485–488.

Caluya, Kenneth F and Abhishek Halder (2020). "Finite horizon density steering for multi-input state feedback linearizable systems". In: *2020 American Control Conference (ACC)*. IEEE, pp. 3577–3582.

Chang, Zinuo et al. (2026). "Efficient iterative proximal variational inference motion planning". In: *Robotics and Autonomous Systems* 197, p. 105267. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2025.105267.

Chen, Gang et al. (2022). "What should be the input: Investigating the environment representations in sim-to-real transfer for navigation tasks". In: *Robotics and Autonomous Systems* 153, p. 104081.

Chen, Wen-Hua et al. (2000). "A nonlinear disturbance observer for robotic manipulators". In: *IEEE Transactions on industrial Electronics* 47.4, pp. 932–938.

Chen, Yongxin, Tryphon T Georgiou, and Michele Pavon (2016a). "On the relation between optimal transport and Schrödinger bridges: A stochastic control viewpoint". In: *Journal of Optimization Theory and Applications* 169, pp. 671–691.

– (2016b). "Optimal transport over a linear dynamical system". In: *IEEE Transactions on Automatic Control* 62.5, pp. 2137–2152.

Chen, Yongxin, Tryphon T Georgiou, and Michele Pavon (2018). "Optimal steering of a linear stochastic system to a final probability distribution, Part III". In: *IEEE Transactions on Automatic Control* 63.9, pp. 3112–3118.

– (2016c). "On the relation between optimal transport and Schrödinger bridges: A stochastic control viewpoint". In: *Journal of Optimization Theory and Applications* 169.2, pp. 671–691.

– (2016d). "Optimal steering of a linear stochastic system to a final probability distribution, Part II". In: *IEEE Transactions on Automatic Control* 61.5, pp. 1170–1180.

Chen., Y., T.T. Georgiou, and M. Pavon (2016). "Optimal steering of a linear stochastic system to a final probability distribution, Part I". In: *IEEE Transactions on Automatic Control* 61.5, pp. 1158–1169.

Clark, Jonathan E et al. (2001). "Biomimetic design and fabrication of a hexapedal running robot". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 4, pp. 3643–3649.

Coleman, David et al. (2014). "Reducing the barrier to entry of complex robotic software: a moveit! case study". In: *arXiv preprint arXiv:1404.3785*.

Collins, Steven H and Andy Ruina (2005). "A bipedal walking robot with efficient and human-like gait". In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1983–1988.

Dellaert, Frank (2021). "Factor graphs: Exploiting structure in robotics". In: *Annual Review of Control, Robotics, and Autonomous Systems* 4.1, pp. 141–166.

Drnach, Luke and Ye Zhao (2021). "Robust trajectory optimization over uncertain terrain with stochastic complementarity". In: *IEEE Robotics and Automation Letters* 6.2, pp. 1168–1175.

Durrant-Whyte, Hugh F (1988). "Uncertain geometry in robotics". In: *IEEE Journal on Robotics and Automation* 4.1, pp. 23–31.

Garage, Willow (2012). *PR2 user manual*.

González, David et al. (2015). "A review of motion planning techniques for automated vehicles". In: *IEEE Transactions on intelligent transportation systems* 17.4, pp. 1135–1145.

Grossman, Robert L et al. (1993). *Hybrid systems*. Vol. 736. Springer.

Haarnoja, Tuomas et al. (2018). "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor". In: *International Conference on Machine Learning*. PMLR, pp. 1861–1870.

Heiss, Florian and Viktor Winschel (2008). "Likelihood approximation by numerical integration on sparse grids". In: *Journal of Econometrics* 144.1, pp. 62–80.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel (2020). "Denoising diffusion probabilistic models". In: *Advances in neural information processing systems* 33, pp. 6840–6851.

Horak, Dan T (1988). "Failure detection in dynamic systems with modeling errors". In: *Journal of Guidance, Control, and Dynamics* 11.6, pp. 508–516.

Hutter, Marco et al. (2011). "Scarleth: Design and control of a planar running robot". In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 562–567.

Johannink, Tobias et al. (2019). "Residual reinforcement learning for robot control". In: *2019 international conference on robotics and automation (ICRA)*. IEEE, pp. 6023–6029.

Johnson, Aaron M, Samuel A Burden, and Daniel E Koditschek (2016). "A hybrid systems model for simple manipulation and self-manipulation systems". In: *The International Journal of Robotics Research* 35.11, pp. 1354–1392.

Kalakrishnan, Mrinal et al. (2011). "STOMP: Stochastic trajectory optimization for motion planning". In: *IEEE international conference on robotics and automation*, pp. 4569–4574.

Kappen, Hilbert J, Vicenç Gómez, and Manfred Opper (2012). "Optimal control as a graphical model inference problem". In: *Machine learning* 87, pp. 159–182.

Kong, Nathan J, Chuanzheng Li, et al. (2023). "Hybrid iLQR model predictive control for contact implicit stabilization on legged robots". In: *IEEE Transactions on Robotics*.

Kong, Nathan J, J Joe Payne, George Council, et al. (2021). "The Salted Kalman Filter: Kalman filtering on hybrid dynamical systems". In: *Automatica* 131, p. 109752.

Kong, Nathan J, J Joe Payne, James Zhu, et al. (2024). "Saltation matrices: The essential tool for linearizing hybrid dynamical systems". In: *Proceedings of the IEEE*.

Krishnamurthy, Vikram (2016). *Partially observed Markov decision processes*. Cambridge university press.

Kuindersma, Scott et al. (2016). "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot". In: *Autonomous robots* 40, pp. 429–455.

Laszlo, Joseph, Michiel van de Panne, and Eugene Fiume (1996). "Limit cycle control and its application to the animation of balancing and walking". In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 155–162.

LaValle, Steven M (2006). *Planning algorithms*. Cambridge university press.

Lee, Yeongseok, Minsu Cho, and Kyung-Soo Kim (2022). "Gpu-parallelized iterative lqr with input constraints for fast collision avoidance of autonomous vehicles". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4797–4804.

Lovejoy, William S (1991). "A survey of algorithmic methods for partially observed Markov decision processes". In: *Annals of Operations Research* 28.1, pp. 47–65.

Lygeros, John et al. (2003). "Dynamical properties of hybrid automata". In: *IEEE Transactions on automatic control* 48.1, pp. 2–17.

Mallick, Tanwi, Partha Pratim Das, and Arun Kumar Majumdar (2014). "Characterizations of noise in Kinect depth images: A review". In: *IEEE Sensors Journal* 14.6, pp. 1731–1740.

Manchester, Ian R, Uwe Mettin, et al. (2011). "Stable dynamic walking over uneven terrain". In: *The International Journal of Robotics Research* 30.3, pp. 265–279.

Manchester, Ian R, Mark M Tobenkin, et al. (2011). "Regions of attraction for hybrid limit cycles of walking robots". In: *IFAC Proceedings Volumes* 44.1, pp. 5801–5806.

Mesbah, Ali (2018). "Stochastic model predictive control with active uncertainty learning: A survey on dual control". In: *Annual Reviews in Control* 45, pp. 107–117.

Mohammadi, Alireza et al. (2013). "Nonlinear disturbance observer design for robotic manipulators". In: *Control Engineering Practice* 21.3, pp. 253–267.

Mukadam, Mustafa, Jing Dong, et al. (2018). "Continuous-time Gaussian process motion planning via probabilistic inference". In: *The International Journal of Robotics Research* 37.11, pp. 1319–1340.

Mukadam, Mustafa, Xinyan Yan, and Byron Boots (2016). "Gaussian process motion planning". In: *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 9–15.

Nguyen, Chuong V, Shahram Izadi, and David Lovell (2012). "Modeling kinect sensor noise for improved 3d reconstruction and tracking". In: *2012 second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*. IEEE, pp. 524–530.

Okamoto, Kazuhide, Maxim Goldshtein, and Panagiotis Tsiotras (2018). "Optimal covariance control for stochastic systems under chance constraints". In: *IEEE Control Systems Letters* 2.2, pp. 266–271.

Okamoto, Kazuhide and Panagiotis Tsiotras (2019). "Optimal stochastic vehicle path planning using covariance steering". In: *IEEE Robotics and Automation Letters* 4.3, pp. 2276–2281.

Plancher, Brian and Scott Kuindersma (2018). "A performance analysis of parallel differential dynamic programming on a gpu". In: *International Workshop on the Algorithmic Foundations of Robotics*. Springer, pp. 656–672.

Quigley, Morgan et al. (2009). "ROS: an open-source Robot Operating System". In: *ICRA workshop on open source software*. Vol. 3. 3.2. Kobe, Japan, p. 5.

Ratliff, Nathan et al. (2009). "CHOMP: Gradient optimization techniques for efficient motion planning". In: *IEEE international conference on robotics and automation*, pp. 489–494.

Ridderhof, Jack, Kazuhide Okamoto, and Panagiotis Tsiotras (2019). "Nonlinear uncertainty control with iterative covariance steering". In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, pp. 3484–3490.

Rooks, Brian (2006). "The harmonious robot". In: *Industrial Robot: An International Journal* 33.2, pp. 125–130.

Schulman, John et al. (2014). "Motion planning with sequential convex optimization and convex collision checking". In: *The International Journal of Robotics Research* 33.9, pp. 1251–1270.

Shental, Ori et al. (2008). "Gaussian belief propagation solver for systems of linear equations". In: *2008 IEEE international symposium on information theory*. IEEE, pp. 1863–1867.

Song, Yang et al. (2021). "Score-based generative modeling through stochastic differential equations". In: *International Conference on Learning Representations*.

Swevers, Jan, Walter Verdonck, and Joris De Schutter (2007). "Dynamic model identification for industrial robots". In: *IEEE control systems magazine* 27.5, pp. 58–71.

Tassa, Yuval (2011). "Stochastic Complementarity for Local Control of Discontinuous Dynamics". In: *Robotics: Science and Systems VI*, p. 169.

Theodorou, Evangelos, Jonas Buchli, and Stefan Schaal (2010). "A generalized path integral control approach to reinforcement learning". In: *The Journal of Machine Learning Research* 11, pp. 3137–3181.

Thrun, Sebastian (2000). "Probabilistic algorithms in robotics". In: *Ai Magazine* 21.4, pp. 93–93.

– (2002). "Probabilistic robotics". In: *Communications of the ACM* 45.3, pp. 52–57.

Todd, David J (2013). *Walking machines: an introduction to legged robots*. Springer Science & Business Media.

Toussaint, Marc (2009). "Robot trajectory optimization using approximate inference". In: *Proceedings of the 26th annual international conference on machine learning*, pp. 1049–1056.

Van Damme, Michaël et al. (2011). "Estimating robot end-effector force from noisy actuator torque measurements". In: *2011 IEEE international conference on robotics and automation*. IEEE, pp. 1108–1113.

Van Erven, Tim and Peter Harremos (2014). "Rényi divergence and Kullback-Leibler divergence". In: *IEEE Transactions on Information Theory* 60.7, pp. 3797–3820.

Westervelt, Eric R et al. (2018). *Feedback control of dynamic bipedal robot locomotion*. CRC press.

Yu, Hongzhe and Yongxin Chen (2023). "A Gaussian Variational Inference Approach to Motion Planning". In: *IEEE Robotics and Automation Letters* 8.5, pp. 2518–2525. DOI: 10.1109/LRA.2023.3256134.

– (2024). *Stochastic Motion Planning as Gaussian Variational Inference: Theory and Algorithms*. arXiv: 2308.14985 [cs.RO]. URL: https://arxiv.org/abs/2308.14985.

Yu, Hongzhe, Zhenyang Chen, and Yongxin Chen (2023). "Covariance steering for nonlinear control-affine systems". In: *arXiv preprint arXiv:2108.09530*.

Yu, Hongzhe, Diana Frias Franco, et al. (2024a). "Optimal Covariance Steering of Linear Stochastic Systems with Hybrid Transitions". In: *arXiv preprint arXiv:2410.13222*.

– (2024b). "Path Integral Control for Hybrid Dynamical Systems". In: *arXiv preprint arXiv:2411.00659*.

Zhang, Wei et al. (2014). "Applications of the cross-entropy method to importance sampling and optimal control of diffusions". In: *SIAM Journal on Scientific Computing* 36.6, A2654–A2672.

Zhao, Rui, Xudong Sun, and Volker Tresp (2019). "Maximum entropy-regularized multi-goal reinforcement learning". In: *International Conference on Machine Learning*. PMLR, pp. 7553–7562.