Data-Driven Safety-Critical Autonomy in Unknown, Unstructured, and Dynamic Environments

Thesis by

Skylar X. Wei

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy, Control and Dynamical Systems

Caltech

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2024
Defended February 6, 2024

ACKNOWLEDGEMENTS

I extend my deepest gratitude to my advisor, Professor Joel W. Burdick, for his unwavering support and guidance throughout my graduate studies. His profound insights into robotic systems have not only shaped my research but his caring and patient mentorship has guided me through various challenges.

My sincere thanks to Professor Aaron Ames, chair of my thesis committee, whose expertise in safety has significantly influenced my work in safety-critical autonomy. I am equally grateful to Professor Richard Murray, my first-year advisor and committee member, whose introduction to diverse research areas in controls and dynamical systems has greatly broadened my perspective. Special appreciation goes to Professor Steven Brunton for his exceptional seminar on data-driven systems, which ignited my passion for physics-informed discrepancy learning. His willingness to share knowledge from an applied mathematics viewpoint has been invaluable.

I am thankful to DARPA, JPL, and TII for funding the LINC, Aerial Manipulation, Axial Descent, and risk-aware drone racing projects. These projects have not only enhanced my multidisciplinary skills but also provided opportunities for collaboration with graduate peers like Sorina Lupu, John Lanthrop, Fengze Xie, and Monica Li, as well as researchers Patrick Spieler, Dr. Marcel Veismann, Dr. James Preiss, Anna Sabel, Dr. Ersin Das, Dr. Matt Anderson, Thomas Touma, Dr. Yashwanth Nakka, Dr. Jonathan Becktor, and mentors including Prof. SoonJo Chung, Dr. Jacob Izraelevitz, Dr. Jeff Delune, Dr. Arash Kalantari, and Dr. Amir Rahmani.

I am grateful for the support and intellectual stimulation provided by my lab and office mates and collaborators Carl Folkestad, Amanda Bouman, Ida Qin, Daniel Pastor, Billy Welch, Stella Kombo, Emily Fourney, Vince Kurtz, Victor Zendejas, Chian Yeh Goh, Sergio Esteban, Tracy Lu, Gilbert Bahati, Prithvi Akella, Jag Boddapati, Julian Humml, Alejandro Stefan, Peder Harderup, Luis Pabon Madrid, and Thomas Maxfield.

Special acknowledgment goes to Anushri Dixit for inspiring my study of finite-time optimal control in a risk framework. I also thank Lu Gan for sharing her expertise in vision and mapping.

My deep appreciation is for my friends Yue Xu, Amy Li, and Wenying Zhu. I will never forget the fun activities and gatherings during this challenging Ph.D. journey.

I am forever grateful to my grandmother Ying for her endless patience, understanding, and emotional support. Her philosophy and perspectives have shaped my life principles and foundations. I also thank my Aunt Summer, Uncle Howard, cousin Kaylie, and the extended Wong family for their nurturing support. My gratitude extends to my parents, my stepfather Carmine, and all my family members. Most importantly, I am profoundly grateful to my partner, Min, for her support and companionship.

ABSTRACT

This thesis addresses the critical challenge of ensuring safety in autonomous exploration within unknown, unstructured, dynamic environments, a domain filled with various types of uncertainties. These include model uncertainties in system dynamics, localization uncertainties stemming from measurement noises, and the risks of collision in environments with dynamic obstacles. Traditional models for vehicle planning and control are often simplified for computational feasibility, but this simplification without careful analysis can compromise safety and system stability. My research introduces a novel, comprehensive framework to provide probabilistically safe planning and control for robot autonomy, structured around three components:

Probabilistic Uncertainty Quantification for Model Mismatches
This segment focuses on identifying model discrepancies given closed-loop tracking data in an unstructured environment where a reduced-order robot model is used for planning and control. The disturbance is modeled as a scalar-valued stochastic process of a norm on the difference between the reduce-order robot model and actual system evolution. In an online and risk-aware framework, Gaussian Process Regression is employed to extract the probabilistic upper bound to such stochastic process, referred to as the *Surface-at-Risk*. Theoretical guarantees on the accuracy of the fitted discrepancy surface are analyzed and verified to the data sets collected during system operation. In an offline setting, conformal prediction, a statistical inference tool, is employed to obtain probabilistic upper bounds of matched and unmatched model disturbance in the system from data, without any assumption of the latent probability distribution governing these discrepancies. Building on these bounds, the robot's nominal ancillary controller is augmented for extending robustness and stability guarantees of the closed-loop system in the face of such discrepancies. Additionally, a *maximum tracking error tube* is constructed along the planned trajectory using the reduced-order model. Such error tubes describe the maximum permissible deviation in actual trajectory tracking under the augmented ancillary controller and the worst-case matched and unmatched model uncertainties, thereby delineating safe operational boundaries for the system.

Data-Driven Unsafe Set Prediction for Dynamic Obstacles
This thesis topic develops an online, data-driven predictive model for dynamic obstacles, accounting for measurement noise and low-frequency data rates. First inspired by singular spectrum analysis (SSA), a time-series forecast technique, obstacle

models characterized by linear recurrence relationships are extracted from real-time position observables. Using the statistical bootstrap technique, a set of predicted obstacle trajectories are constructed, which in turn are reformulated into deterministic distributionally robust obstacle avoidance constraints, reflecting a user-defined risk tolerance. Further refining the obstacle predictor for intention-unknown obstacles, a linear, time-varying model is learned from data using time-delay embedding of obstacle position observables. Additive process and measurement noises are anticipated in the learned model, where their intensities are estimated from data. For inferring prediction uncertainties, a companion data-driven Kalman Filter (DDKF) is constructed to forecast obstacle positions and uncertainties. This "heuristic unsafe set" from DDKF is then dynamically calibrated using adaptive conformal prediction, ensuring safety without relying on any distribution assumptions regarding the uncertainties or model accuracy. The calibrated sets, called conformal prediction sets, are then reformulated into convex state constraints.

Safety-Critical Planning

The thesis proposes two methods for ensuring safety in planning and navigation: Probabilistic-Safe Model Predictive Control (MPC) and Probabilistic-Safe Model Predictive Path Integral (MPPI) given uncertainties arising from operating in unknown, unstructured, and dynamic environments. The MPC approach integrates the quantified obstacle avoidance constraints into a convex program to balance computational tractability while providing probabilistic safety guarantees. In contrast, the MPPI method, a sampling-based strategy, incorporating unsafe sets into a cost map derived from sensory data, optimizes reference tracking trajectory while guaranteeing collision avoidance up to a user-defined risk tolerance.

In unknown and cluttered environments automatically, the proposed framework learns an upper bound on model residuals from data and systematically calculates the safety buffers needed to provide the desired probabilistic safe navigation of robotics systems. Additionally, in the presence of dynamic obstacles, the proposed data-driven predictor systematically extracts an obstacle model and makes obstacle-occupied unsafe set forecasts. These features largely eliminate the "hand tuning" of the underlying planner and controller that is normally required in heuristic-based algorithms. The efficacy of these proposed frameworks is empirically validated through Monte Carlo Simulations, alongside hardware validations on both ground and aerial vehicles, demonstrating their robustness, versatility, and applicability in real-world scenarios.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] Wei, Skylar X.*, Dixit, Anushri*, Tomar, Shashank, and Burdick, Joel W., "Moving obstacle avoidance: A data-driven risk-aware approach," *IEEE Control Systems Letters*, vol. 7, pp. 289–294, 2023. doi: `10.1109/LCSYS.2022.3181191`,
Contribution: SSA predictor and bootstrap MPC planner, simulation implementation, and validation.

[2] Wei, Skylar X., Gan, Lu, and Burdick, Joel W., "A safety-critical framework for ugvs in complex environment: A data-driven discrepancy-aware approach," in *Preprint submitted to Robotics and Autonomous Systems*, Elsevier, 2024,
Contribution: theory, hardware implementation, and validation.

[3] Dixit, Anushri*, Lindemann, Lars*, Wei, Skylar X., Cleaveland, Matthew, Pappas, George J., and Burdick, Joel W., "Adaptive conformal prediction for motion planning among dynamic agents," in *Learning for Dynamics and Control Conference*, PMLR, 2023, pp. 300–314,
Contribution: constructing the obstacle predictor, simulation implementation, and validation.

[4] Akella, Prithvi, Wei, Skylar X., Burdick, Joel W., and Ames, Aaron, "Learning disturbances online for risk-aware control: Risk-aware flight with less than one minute of data," in *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, ser. Proceedings of Machine Learning Research, vol. 211, PMLR, 15–16 Jun 2023, pp. 665–678,
Contribution: hardware construction, implementation, and validation.

[5] Wei, Skylar X., Dixit, Anushri, Brunton, Steven L., and Burdick, Joel W., "Data-driven moving obstacle avoidance: A planning framework with probabilistic safety guarantees," in *In preparation for The International Journal of Robotics Research*, Sage, 2024,
Contribution: theory, simulation implementation, hardware implementation, and validation.

[6] Folkestad, Carl, Wei, Skylar X., and Burdick, Joel W., "Koopnet: Joint learning of koopman bilinear models and function dictionaries with application to quadrotor trajectory tracking," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 1344–1350,
Contribution: hardware validation.

*Authors contributed equally.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Chapter 1

INTRODUCTION

In recent decades, the field of robotics has garnered significant public interest as it transitions from factory and laboratory settings into more urban, human-centric environments. Recent applications range from autonomous vehicles offering taxi services to robot vacuums that clean floors, kitchen manipulators preparing food, and exoskeleton robots enhancing or rehabilitating human movement. These innovations, particularly relevant in cluttered, uncertain, and potentially challenging environments, highlight the crucial role of online path planning, integrated with obstacle behavior classification and avoidance [1].

The motion planning of autonomous systems in dynamic environments necessitates an acute awareness of environment-induced uncertainties. For example, a self-driving car must predict the erratic movement of other vehicles, and a mobile robot in a crowded space must assess the uncertainties of pedestrian movements. These applications are safety-critical, given the unknown intentions of surrounding agents and unstructured environments. Robotic systems must be adept at planning reactive behaviors. Confidence-based motion planning is a leading approach in this domain, integrating high-level path planning and data-driven model refinement based on robot-environment interactions [2]–[4], along with robust ancillary controllers for disturbance rejection.

Moreover, safety must also be enforced in environments that are unknown and unstructured. This concern extends across various fields, including agriculture [5], search and rescue operations [6], package delivery [7], and mining [8]. Challenges in these sectors often stem from incomplete or unknown terrain and obstacle data [9], [10] and model mismatches due to the inherent unstructured nature of these environments [11], [12]. For instance, a self-driving car must navigate effectively on icy roads or rough terrain where its trajectory tracking accuracy is significantly compromised, and aircraft autopilots need to stabilize themselves against random wind gusts and weathering to provide safe and comfortable flights. Similarly, search and rescue missions typically occur in unknown settings where robots must intelligently discern traversal paths.

Given the intricacies presented by these environments, the creation of systematic and verifiable probabilistic frameworks for robot traversal becomes imperative. Establishing such frameworks is crucial for ensuring safety in scenarios characterized by unknown, unstructured, and dynamic conditions. This thesis categorizes these scenarios into two primary classes, each underscoring a distinct aspect of robotic autonomy: (1) Unknown and Unstructured Autonomy (UUA), focusing on navigating environments where terrain and obstacle characteristics are not predefined, and (2) Moving Obstacle Avoidance (MOA), emphasizing the dynamic aspect of environments where moving entities with unknown intentions, such as pedestrians or other vehicles, pose variable challenges to robot autonomy.

This chapter was adapted from: [13] [14], [15], [16], [17].

1.1 Motivations

This thesis centers around synthesizing planning and control frameworks for robotic autonomy when facing unknown, unstructured, and dynamic environments. Ensuring system stability and safety in the presence of these uncertainties necessitates robust vehicle planning and control strategies. Addressing these challenges, my research has established a comprehensive framework by answering the following key questions:

Q1 How to quantify uncertainty arising from model mismatches?

Q2 What is the impact of unstructured uncertainties on system performance and stability?

Q3 What methods can be employed to predict dynamic obstacles' future trajectories?

Q4 How to reason about prediction accuracy given a probabilistic risk tolerance?

Q5 How to synthesize planners and controllers, considering the quantified uncertainties and models, to ensure probabilistic safety and stability for robot autonomy?

1.2 Literature Review

Disturbance Modeling and Robust Planning

Prior research has studied the impact of unmodeled disturbances on vehicle planning and control using frameworks such as input-to-state safe control [18], [19] and risk-aware control [20], [21]. However, these methods often assume *a priori* model knowledge or require a minimum understanding of the disturbances' magnitudes or distributions. Bayesian Optimization and Reinforcement Learning methods can bypass the uncertainty or model identification steps, directly learning risk-aware control policies in a model-free fashion [22], [23]. These approaches still come with the assumption of *a priori* knowledge of the disturbances, or that they can be sampled. More recently, the union of Neural Networks with adaptive control [24] has demonstrated remarkable tracking improvements in drone control given unknown residual dynamics. However, the theoretical robustness guarantees in [24] rely on knowledge of wind disturbance upper bounds, and they do not consider obstacles or actuation constraints.

A learned control policy can be combined with an optimal path planner, as demonstrated in robust model predictive control [25], [26] and chance-constrained stochastic optimal control [2], [27]. Although these methods construct a deterministic problem surrogate to the original probabilistic one, they often require constraint convexification, such as sequential constraint linearization around fixed points, and optimality can only be reached in infinite sequential iterations. For instance, Monte Carlo Sample propagation [3], [28] and scenario-based approaches [29] can require a very large number of samples to reach the desired obstacle avoidance probability guarantees.

The sampling-based Model Predictive Path Integral control (MPPI) method has proven versatility in off-road racing applications [30]. As a model-based strategy, MPPI may exhibit suboptimal performance in the face of modeling uncertainties and disturbances. Tube-based [31] and robust MPPI variants [32] have been proposed to address these challenges. Both of these methods integrate an ancillary controller, such as iterative Linear Quadratic Gaussian (iLQG), to improve tracking performance with measurement feedback and robustness to uncertainties. These methods' assumptions on sequential linearization and the additive process and measurement noise being Gaussian limits their applicability to nonlinear systems with generalized additive model uncertainties.

In unknown environments, robot navigation often relies on sensor-based occupancy grid maps [33], [34], which are commonly used for global path planning [35]. Grid maps can be used for local optimization-based path planning by converting the maps into signed distance functions [36], [37], which can be locally convexified to serve as the collision-avoiding state constraints. Alternatively, occupancy-based risk maps have been used for sampling-based local planners [9], [38]–[40]. To alleviate the computation and convexity burden, the MPPI algorithm allows direct grid-map-based assessment for trajectory costs [40]. However, systematic parameter and cost tuning is required to avoid the algorithm making undesirable decisions [41], [42]. Lastly, direct estimation of traversability from sensors is also a popular technique for robotic navigation in complex environments leveraging expert heuristics [9], [43] or self-supervised learning [44]. However, it is not straightforward to extend these hardware successes to provide provable safety guarantees.

Obstacle Avoidance

This overview covers the diverse and complex strategies for dynamic obstacle avoidance, highlighting the challenges and opportunities in developing a novel framework with safety guarantees with minimal assumptions about obstacle behaviors and data access. Dynamic obstacle avoidance has garnered substantial attention in research because it plays an essential role in field robotics and autonomy, including urban autonomous driving, defense applications, and robot delivery systems [45], [46]. From dynamic obstacle modeling and behavior characterization to control and planning policy synthesis, there are numerous existing approaches to finding collision-free safe maneuvers. The related works are reviewed under the following criteria:

1. The obstacle's type, behavior, intent, and quantity are unspecified.

2. Obstacle position measurements, obtained at a fixed rate of approximately $20-30$ Hz using LiDAR sensors, are the sole source of information. These measurements are subject to corruption by Gaussian White Noise, with an unknown covariance matrix.

3. Provable Safety Guarantees are provided with Hardware Validations.

Obstacle model-free approaches like artificial potential fields (APF) and relative velocity methods are common. For instance, Lam *et. al.* [47] apply APF with stochastic reachable sets in Human-Centered environments. Or, one can plan the agent's path offline using a Probabilistic Roadmap (PRM) in a field of static obstacles and then replan when dynamical behaviors are observed [48]. However, lacking prior knowledge of the obstacle behaviors can lead to overly conservative (worst-case-based) safe maneuvers.

Control barrier functions (CBFs), another popular solution to obstacle avoidance problems, phrase collision avoidance as a controller synthesis problem [49]–[51]. The collision-preventing safety constraints are typically affine with respect to the control input, enabling the use of convex optimization for safe and optimal input synthesis. Nevertheless, major theoretical advancement in CBFs for obstacle avoidance often presumes perfect knowledge of the robot and obstacle dynamics or rely on worst-case bounds that can be overly conservative [52]–[54]. The recent surge of backup CBF strategies under torque constraints [55] has shown promise in multi-agent obstacle avoidance settings. However, in situations where the dynamics, control, and intentions of the obstacle remain unknown, CBF-based strategies require accurate obstacle trajectory or rates estimate. [56] ingeniously leveraged velocity cones and CBFs to decouple the control of linear speed and steering in ground vehicles to produce effective obstacle avoidance strategies. Yet, this approach uses a fixed upper bound on the dynamics of all moving obstacles to provide robustness which can be excessively cautious, especially when obstacles move at different speeds.

For UAVs, Tordesillas and How presented a state-of-the-art 3D decentralized trajectory planner, which has shown robustness against communication delays and efficacy in dynamic environments and has been validated on hardware [57], [58]. These planners achieve real-time dynamic obstacle avoidance by representing their trajectories each with an outer polyhedral convex hall, demonstrating computational superiority and efficiency in scenarios with numerous obstacles. Nonetheless, these planners require the establishment of agent-to-agent communications in multi-agent obstacle avoidance or an accurate obstacle trajectory predictor for extrapolating obstacle future behaviors. The trajectory predictor is polynomial extrapolation with a 3D Gaussian position uncertainty [59]. However, the accuracy of such extrapolation can be compromised in the presence of measurement noise and limited real-time data, leading to potential collision risks. Such predictions may be of limited value in the presence of uncooperative obstacle agents.

Chance-constrained Model Predictive Control (CCMPC) addresses obstacle uncertainties through chance constraints, such as those related to position uncertainties [60]–[63], and has been hardware-validated with perception-based obstacle detection and localization [64]. Wakabayashi *et. al.* have further enhanced CCMPC by incorporating velocity uncertainties, thereby improving noise resistance. However, the simplification of obstacle dynamics to chained integrator models in these approaches could result in suboptimal predictions of obstacle behavior and overly conservative maneuvers.

Extracting a dynamics model from data is challenging [65], especially when the available data is limited, noisy, and partial. To tackle partial measurements, Takens' embedding theorem [66]) offers a solution that uses partial observations to produce an attractor that is diffeomorphic to the full-state attractor. Koopman-inspired model identification methods [67]–[69] have shown success in obtaining linear time-invariant models from time-series data using time-delay embedding. Yet, it is not evident how to derive an obstacle model from real-time obstacle position measurements as well as quantify the learned model accuracy in the presence of measurement noises.

Recent research introduces prediction monitors [70], [71] which employ conformal prediction to provide one-step guarantees on false negative rates of predictors, which are useful in synthesizing point-wise obstacle-avoiding control inputs.

When offline training datasets are available, a variety of methods have been proposed to learn high-fidelity obstacle dynamics models for pedestrians [72], [73]. Intent-driven models for planning among human agents have estimated agent uncertainty using Bayesian inference [74]–[77]. Additionally, constructing offline motion libraries and performing reachability-set analysis [78], [79] can enrich the set of available motion primitives for identifying obstacle-avoiding maneuvers or assist online planners in synthesizing safe trajectories. Switching-based planning methods, which detect and classify dynamic obstacle behavior against a set of trajectories, such as constant speed, linear, and projectile-like motions [80], [81], require distinguishable obstacle behaviors and the knowledge about the classes of obstacle behaviors. Data-driven trajectory predictors can provide mean and variance information of the predictions, which can be approximated as a Gaussian distribution [82] and used within stochastic planning frameworks [83]–[85]. However, under the criteria defined earlier, these methods require large training sets with ground truth for model calibrations. This characteristic makes it difficult to translate this method to real-time obstacle avoidance. Furthermore, these approaches quantify prediction uncertainty in a heuristic manner, such as making assumptions about the prediction algorithms, and agent models, and their uncertainty distribution is Gaussian.

1.3 Summary of Contribution

The core elements of the proposed data-driven safety-critical autonomy framework are data-driven model identification, uncertainty quantification of model prediction accuracy, and the transformation of risk-aware chance constraints into deterministic state constraints for optimization. The individual contributions presented in this thesis are summarized as follows:

Model Uncertainty Identification and Quantification - Online Framework: Chapter 3 details a risk-aware model augmentation approach via learning disturbance models online that does not require *a priori* disturbance knowledge. The approach is sample-efficient, where the method requires less than a minute of flight data to make risk-aware control improvements on a drone mid-flight. By treating the norm discrepancy between the reduce-order model and true system evolution as a scalar stochastic process, this method proposed to learn a *Surface-at-Risk* using Gaussian Process Regression which provides theoretical and experimentally verifiable probabilistic guarantees. Hence, augmenting the controller with the learned disturbance bounds yields an efficient risk-aware controller.

Model Uncertainty Identification and Quantification - Offline Framework: Chapter 4 introduces a *discrepancy-aware* planning and control framework for the safe traversal of unmanned ground vehicles (UGVs). Specifically, there are unknown model discrepancies between the simplified nominal model used for planning and the true vehicle dynamics. The proposed method augments the vehicle's controller using learned discrepancies given offline training data sets and applies rigorous uncertainty quantification to provide provably safe operation. Example model discrepancies include wheel slipping and skidding, actuation dynamics and delays, and communication delays. The proposed framework automatically learns an upper bound on model residuals from data and systematically calculates the collision buffers needed to provide the desired probabilistic safe traversal of robots in cluttered and unstructured environments.

The proposed framework consists of the following key contributions:

- *Data-Driven Discrepancy Identification*: Using conformal prediction ([86]), a probabilistic upper bound of matched (controllable input) discrepancies and unmatched (uncontrollable drift) discrepancies are identified from training data, without any assumptions on the discrepancy distributions.

- *Controller Augmentation*: The identified upper bounds are then used to augment the vehicle's nominal ancillary controller to ensure closed-loop stabilizability and robustness against uncertainties.

- *Collision Buffer Construction*: To ensure safety given model uncertainties, a maximum trajectory tracking deviation of the closed-loop system is deduced under the augmented controller. This *collision buffer* delineates the boundaries within which the system can operate safely, despite mismatches between the nominal and true vehicle models.

- *Discrepancy-Aware Planner*: A *discrepancy-aware cost map* is constructed from the identified collision buffer and a sensory-derived occupancy map. This cost map can be seamlessly used with Model Predictive Path Integral (MPPI) to generate optimal finite-horizon trajectories that provably adhere to user-chosen risk tolerance.

The effectiveness of the framework is experimentally validated for autonomous high-speed trajectory tracking in a cluttered environment with four different vehicle-terrain configurations. This chapter also showcases the framework's versatility by reformulating it as a driver-assist program, providing collision avoidance corrections based on user joystick commands.

Moving Obstacle Avoidance - Bootstrapping Forecasting: As a proof-of-concept to merge data-driven obstacle trajectory forecast and receding horizon planning, a data-driven moving obstacle avoidance framework is developed and presented in Chapter 5. Such a framework consists of the following steps: (1) discovering the dynamics of a priori unknown moving obstacles, (2) forecasting their trajectories, and (3) providing risk-aware optimal avoidance strategies. This framework replaces the need for obstacle trajectory/model classification while allowing online computation.

Singular Spectrum Analysis (SSA) ([87], [88]) is employed to separate noise from the underlying signal and to extract a predictive model of obstacle behavior. Inspired by Fasel *et. al.* [89], a set of obstacle trajectory predictions are made using bootstrapping. An MPC planner then incorporates the set of obstacle forecasts as an affine conservative approximation of a distributionally robust chance constraint (DRCC). This constraint is then efficiently recast in a risk-aware manner, allowing an MPC optimization based on sequential convex programming [90], [91].

Monte Carlo simulations verify the planner's ability to uphold the user-chosen chance constraint. The risk-aware reformulation not only gives provable probabilistic collision avoidance guarantees but also allows an online execution of the planner.

Moving Obstacle avoidance - Probabilistic Guarantees: To provide probabilistic obstacle avoidance guarantees, a refined framework with rigorous prediction uncertainty quantification is developed, detailed in Chapters 6 and 7. This novel framework consists of three main components: (1) a Data-Driven Horizon Predictor, (2) Heuristic and Conformal Unsafe Sets, and (3) the Prediction-Aware Planner. An optimization-based model predictive planner is formulated that tactically incorporates deterministic predicted unsafe sets constructed from rigorous uncertainty quantification.

(1) Data-Driven Horizon Predictor: Using past noisy obstacle position measurement, a data-driven model learning framework is built using ideas from time-delay embedding and Koopman operators. Specifically, this method obtains a lifted linear, time-varying input-output model with additive noises for characterizing obstacle dynamics from data. Obstacle trajectory future predictions can be obtained by forward-propagating the learned model. By employing an optimal hard thresholding technique, the proposed algorithm effectively separates the desired signal from measurement noises, thereby improving the prediction accuracy and robustness to noise corruption.

(2) Heuristic and Conformal Unsafe Sets: A pairing Data-Driven Kalman Filter is proposed for the learned obstacle model to make unsafe set predictions for future time instances given a user-defined risk tolerance. Initially considered as heuristic due to the Gaussian assumption of noise estimations, this limitation is lifted by using adaptive conformal prediction to calibrate the prediction uncertainty rigorously. The proposed conformal unsafe set algorithm offers provable probabilistic coverage without relying on assumptions about obstacle behaviors or noise distributions.

(3) Prediction-Aware Planner: For practical implementation, the probabilistic heuristic and conformal unsafe sets are reformulated into sequentially affine state constraints. This reformulation casts the chance-constrained moving obstacle avoidance problem into a deterministic convex optimal control. Most importantly, any feasible solution to the proposed prediction-aware planner is theoretically proven to ensure probabilistic obstacle avoidance on average.

To validate the efficacy of this framework, extensive Monte Carl simulations are conducted to scrutinize the framework's ability to navigate through obstacles exhibiting various behaviors, all while adhering to a predefined risk tolerance. Further, real-world experiments using two distinct hardware platforms are performed to attest to the proposed framework's practical utility. These experiments showcase a drone successfully dodging a moving obstacle in confined spaces and a differential-driven ground rover tracking predefined waypoints while circumventing multiple obstacles with unknown intentions.

1.4 Thesis Structure and Organization

Organization

The thesis adheres to the following notational convention for clarity and precision. Scalar quantities are denoted using standard, lowercase letters, while vector quantities are represented in boldface lowercase. Capital letters are reserved for matrices, and Greek letters may also be utilized. Sets and spaces are distinctly indicated using calligraphic and blackboard bold capitalized alphabets.

The set of integers, positive integers, natural numbers, real numbers, positive reals, and non-negative reals are denoted as $\mathbb{Z}, \mathbb{Z}_{>0}, \mathbb{N}, \mathbb{R}, \mathbb{R}_{>0}$, and $\mathbb{R}_{\geq 0}$, respectively. The sequence of consecutive integers, such as $\{i, \cdots, i+k\}$, is denoted as $\mathbb{Z}_i^{i+k}$. A finite sequence of scalars a, $\{a_1, \cdots, a_k\}$, is represented as $\{a_i\}_{i=1}^k$ and for vectors of $\mathbf{a}$ as $\{\mathbf{a}_i\}_{i=1}^k$. General indicators are defined as $\mathbf{1}_{b(a)} : \mathbb{R} \rightarrow \{0, 1\}$ which returns 1 if $b(a)$ is true, and 0 otherwise.

Time dependent variables use a subscript $(\cdot)_t$ to abbreviate $(\cdot)(t)$, and all time dependent variables are causal, i.e. $t \in \mathbb{Z}_{\geq 0}$. To reflect random variable dependencies, subscript $x_{t|t-a:t-b} \triangleq x(t|t-a : t-b)$ denotes that the variable $x(t)$ depends on the time sequence $\{x(k)\}_{k=t-a}^{t-b}$ where $a, b \in \mathbb{Z}_{>0}$ and $a > b$. The parameters that depend on the available data captured after $t = 0$ are abbreviated as $x_{t+1|t} = x_{t+1|0:t}$.

Structure

Chapter 2 provides an in-depth review of the foundational theories underpinning this thesis. Subsequent chapters, 3 and 4, present two complementary frameworks addressing robot autonomy in unstructured environments with unknown model uncertainties. In an online setting, Chapter 3 proposes a method to efficiently learn *a priori* unknown disturbances that a system faces during operation. It presents the *Surface-at-Risk*, a scalar-valued stochastic process derived from the norm difference between the theoretical system model and its actual evolution. Gaussian Process

Regression is employed to establish a probabilistic upper bound for the *Surface-at-Risk*, enhancing controller performance through augmentation. Chapter 4 delves into probabilistic safety guarantees for unmanned ground vehicles traversing unknown and unstructured terrains. Leveraging offline data, a robust model predictive planner coupled with an ancillary controller is proposed. This framework ensures collision-free navigation and robustness against matched and unmatched model uncertainties, identified using conformal prediction techniques without presuming the underlying probability distribution.

Chapters 5-7 focus on safety-critical robot autonomy in environments with dynamic and unknown obstacles. Chapter 5 introduces a data-driven, risk-aware framework for moving obstacle avoidance by predicting the paths of moving obstacles and avoiding them using a model predictive control (MPC) scheme. A bootstrapping technique is employed to predict a set of obstacle trajectories from linear obstacle predictors extracted from data. The bootstrapped predictions are incorporated in the MPC optimization using a risk-aware distributionally robust formulation to provide probabilistic guarantees on obstacle avoidance. Chapter 6, building on the preceding chapter, formalizes and extends the process of acquiring prediction uncertainty using the statistical inferences technique adaptive conformal prediction. Leveraging adaptive conformal prediction, a rigorous technique is proposed to dynamically quantify the prediction uncertainty of any model predictors from an online data stream. Particularly, probabilistic coverage is made for multistep-ahead prediction which is used within an MPC to safely navigate among moving obstacles. Chapter 7 extends the ideas from the previous chapters by providing a refined data-driven technique inspired by Koopman operators that learn dynamic obstacle models with unknown intentions. Relying solely on noisy obstacle position measurements, this finalized framework guarantees probabilistic safety, by deducing an obstacle model from data, forecasting future obstacle-occupied unsafe sets, performing rigorous uncertainty quantification of the predictors, and planning trajectories that ensure collision-free traversal.

Finally, Chapter 8 revisits the thesis contributions by addressing the initial motivational questions presented in Chapter 1. It concludes with a discussion of future research directions to refine the proposed frameworks for achieving safety-critical robot autonomy in complex environments.

Chapter 2

PRELIMINARIES

This chapter reviews theoretical foundations that are used in the remainder of the thesis. A comprehensive overview of several critical building blocks is provided, each playing an important role in the development of my frameworks.

The review begins by exploring the Koopman Operator and the concept of Time-Delay Embedding. These data-driven techniques are instrumental in the identification of dynamic obstacle models, enabling complex behaviors to be extracted and predicted from limited observational data. The chapter next summarizes the conformal prediction method and its variants. These statistical tools are essential for calibrating the uncertainty in data-driven models. Finally, a discussion on the finite-time optimal control problem is provided, focusing on three variants: sequential convex programming for nonlinear model predictive control, the model predictive path integral approach for general nonlinear control problems, and robust model predictive control in the face of additive model uncertainties. These methodologies are crucial for synthesizing trajectories that ensure probabilistic safe autonomy.

This chapter was adapted from: [13] [14], [92], [15]. [16], [17].

2.1 Koopman Operator

Koopman operator theory is a mathematical framework to transform finite-dimensional nonlinear state-space systems into infinite-dimensional linear systems in a Hilbert space of scalar observables (functions on the state) [67], [93]–[95]. Since linear dynamical systems are well-characterized and understood, Koopman-inspired model identification has become a popular method in the field of fluid mechanics [96] and neuroscience [97] to identify a linear dynamic model from observables. A brief review of the Koopman operator is provided in the context of data-driven model reduction. More extensive introductions can be found in [95], [98].

Consider the autonomous, possibly nonlinear, dynamical system

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}), \tag{2.1}$$

with state $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^{n_x}$, and the function $f : \mathcal{X} \to \mathcal{X}$ is assumed to be Lipschitz continuous on $\mathcal{X}$. Denote the flow of the system given an initial state $\boldsymbol{x}_0 \triangleq \boldsymbol{x}(0)$ as $\phi_t(\boldsymbol{x}_0)$ where $\frac{d}{dt}\phi_t(\boldsymbol{x}_0) = \mathbf{f}(\phi_t(\boldsymbol{x}_0))$ for all $\boldsymbol{x}_0 \in \mathcal{X}, t \geq 0$.

The *Koopman operator* K_t where $t \geq 0$ is defined as:

$$K_t\varphi = \varphi \circ \phi_t, \tag{2.2}$$

for all $\varphi \in C(\mathcal{X})$, where $C(\mathcal{X})$ is the space of continuous observables $\varphi : \mathcal{X} \to \mathbb{C}$, and $\circ$ denotes function composition where

$$K_t\varphi(\boldsymbol{x}_0) = \varphi(\phi_t(\boldsymbol{x}_0)) = \varphi(\boldsymbol{x}(t)). \tag{2.3}$$

Most importantly, each $K_t : C(\mathcal{X}) \to C(\mathcal{X})$ is a *linear* operator, which is equivalently the state transition matrix over the space of the observable functions of the nonlinear system (2.1).

An *eigenfunction* of the Koopman operator associated to an eigenvalue $\lambda \in \mathbb{C}$ is any function $\varphi \in C(\mathcal{X})$ such that

$$(K_t\varphi)(\boldsymbol{x}_0) = \varphi(\phi_t(\boldsymbol{x}_0)) = e^{\lambda t}\varphi(\boldsymbol{x}_0). \tag{2.4}$$

The Koopman eigenfunctions, a special type of observable, evolve linearly in time, and therefore they can be used as a basis for the space of observables. Suppose the Koopman operator K_t has a pure point spectrum [99], the dynamics of any vector-valued observable function g can be decomposed as follows

$$g(\boldsymbol{x}(t)) = K_t g(\boldsymbol{x}_0) = \sum_{i=0}^{\infty} \lambda_i \varphi_i(\boldsymbol{x}_0) e^{\lambda_i t} \tag{2.5}$$

which is known as the Koopman mode decomposition [95] which can be infinite in dimension.

Similar to these works [100], [101], it is desirable to obtain a finite-dimensional approximation of the Koopman operator to practically model and identify a system given data sets derived from that system. Specifically, the finite-dimension truncation error can be viewed as a stochastic model uncertainty as

$$g(\boldsymbol{x}(t)) = \tilde{K}_t g(\boldsymbol{x}_0) + \epsilon_g(t) = \underbrace{\sum_{i=0}^{n_\lambda - 1} \lambda_i \varphi_i(\boldsymbol{x}_0) e^{\lambda_t} t}_{\tilde{K}_t g(\boldsymbol{x}_0)} + \underbrace{\sum_{i=n_\lambda}^{\infty} \lambda_i \varphi_i(\boldsymbol{x}_0) e^{\lambda_t} t}_{\epsilon_g(t)}, \tag{2.6}$$

where $\{\varphi_i\}_{i=0}^{n_\lambda - 1}$ are the n_λ data-driven Koopman eigenvectors that correspond to the n_λ largest of data-driven Koopman eigenvalues, denoted as $\{\lambda_i\}_{i=0}^{n_\lambda - 1}$. Explicitly, the finite eigenvalue-eigenvector truncation error is aggregated in the term $\epsilon_g(t)$ as model mismatches.

2.2 Time-Delay Embedding

Time-delay embedding is a popular technique from system identification [102] literature and the time series forecasting community [87] to characterize dynamical systems given partial measurements. Partial measurements are used to reconstruct a representation of the partially observed high-dimensional system by extending the present measurement with a time history of its previous measurements. More concretely, Takens' method of delays embedding [66] is shown to reconstruct qualitative features of the full-state phase-space from delayed partial observations.

For this thesis, time-delay embedding is leveraged for obstacle modeling for moving obstacles with unknown intentions and partial observations (position measurements only). Let $\hat{x}$ be the x-position measurement of an obstacle. Define an L-delay embedding $\boldsymbol{h}_t^{\hat{x},L} \in \mathbb{R}^L$ for such measurement at time t as

$$\boldsymbol{h}_t^{\hat{x},L} = \begin{bmatrix} \hat{x}_{t-L+1} & \hat{x}_{t-L+2} & \cdots & \hat{x}_L \end{bmatrix}^T \tag{2.7}$$

which is comprised of the most recent L measurements concatenated based on its acquired time. In another word, the vector $\boldsymbol{h}_t^{\hat{x},L}$ encodes the x-position trajectory of the obstacle in the time interval $[t - L + 1, t]$.

By horizontally stacking the L-delay embedding vector, a L-embedding Hankel matrix $H_t^{\hat{x},L,N} \in \mathbb{R}^{L\times N}$ is contructed for position measurement $\hat{x}$ as:

$$H_t^{\hat{x},L,N} = \begin{bmatrix} \boldsymbol{h}_{t-N+1}^{\hat{x},L} & \boldsymbol{h}_{t-N+2}^{\hat{x},L} & \cdots & \boldsymbol{h}_t^{\hat{x},L} \end{bmatrix} \tag{2.8}$$

where N is the total time window in which the Hankel matrix is constructed that satisfies the inequality $t \geq N >> L$. The repeating patterns in the Hankel matrix represent underlying trends and oscillations, which can be extracted from its covariance matrix: $X_t^{\hat{x}} = H_t^{\hat{x},L,N}\left(H_t^{\hat{x},L,N}\right)^T$. More importantly, the singular value decomposition of the Hankel matrix can produce A (state-transition), B (actuation), and C (observation) matrices in state-space realization, a popular technique in classical state-space system identification [103].

2.3 Conformal Prediction

Conformal prediction provides model- and assumption-free uncertainty quantification to black-box models [104], [105]. As illustrated in Fig. 4.3, conformal prediction can produce set guarantees to contain the desired ground truth on any pre-trained model upholding a user-specified confidence level.

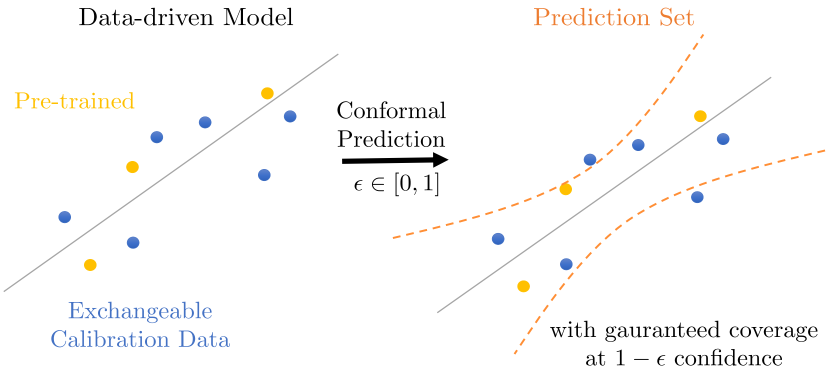

Figure 2.1: Conformal Prediction schematics adapted from [105] that illustrates its application in producing a prediction set of a data-driven model based on a specified confidence $1 - \epsilon$.

Definition 1. *([106]) The random variable $\{Y_0, Y_1, \ldots, Y_n\}$ are exchangeable if the joint probability distribution of the random vector $[Y_0, Y_1, \ldots, Y_n]$ is the same as the joint probability distribution of $[Y_{\pi_0}, Y_{\pi_1}, \ldots, Y_{\pi_n}]$ for any permutation $\{\pi_0, \pi_1, \ldots, \pi_n\}$ of the indices $\{0, 1, \ldots, n\}$.*

Let $Y_0, \ldots, Y_{n-1}$ be n exchangeable random variables where $Y_i \in \mathbb{R}$ is a *nonconformity score*. The nonconformity score is typically chosen to express the difference between the pre-training model of an unknown system and calibration data obtained from the unknown system. Using a prediction model of a black box system as an example. Example nonconformity scores Y_i can be the difference between the predicted outcomes and the true outcomes of sample inputs to the black box system. The conformal prediction algorithm uses these nonconformity scores to calibrate the model's prediction regions. Roughly speaking, the prediction region can be viewed as an uncertainty set along the predicted output where the size of the set is correlated with the desired confidence $(1 - \epsilon) \in [0, 1]$ guarantees. In particular, the conformal prediction algorithm outputs a value $\overline{Y}_\epsilon \in \mathbb{R}_{\geq 0}$ measuring the size of the prediction region such that

$$\mathbb{P}[Y \leq \overline{Y}_\epsilon] \geq 1 - \epsilon$$

given $\epsilon \in (0, 1)$.

In the context of dynamical systems with uncertainties, consider the following autonomous system with system state $\boldsymbol{x}$:

$$\boldsymbol{x}_{k+1} = \hat{f}(\boldsymbol{x}_k) = f(\boldsymbol{x}_k) + \tilde{f}(\boldsymbol{x}_k) \tag{2.9}$$

where the functions $\hat{f}$, f, and $\tilde{f}$ are the true system model, nominal system model, and model discrepancy between the nominal and the true system, respectively. In this case, the norm difference between the true, $\hat{f}(\boldsymbol{x}_k)$ and nominal $f(\boldsymbol{x}_k)$ model evolution given initial condition $\boldsymbol{x}_k$ over time Δt measures the discrepancy magnitude given sample state $\boldsymbol{x}_k$. By selecting $Y_i \triangleq \|\hat{f}(\boldsymbol{x}_k) - f(\boldsymbol{x}_k)\| = \|\tilde{f}(\boldsymbol{x}_k)\|$ as the nonconformity score, a large Y^i can be interpreted as a large model discrepancy, indicating that there is a from poor matching between the nominal and true model. In the context of making obstacle position predictions for time $t+k$ given current time t, one can use the norm difference between the prediction made at time t, denoted as $\tilde{\boldsymbol{p}}_{t+k|t}$, and the measured obstacle position at time $t+k$, denoted as $\hat{\boldsymbol{p}}_{t+k}$, as the nonconformity score, *i.e.*, $Y_i = \|\tilde{\boldsymbol{p}}_{t+k|t} - \hat{\boldsymbol{p}}_{t+k}\|$. Similarly, a large nonconformity score means a large prediction error which requires a large prediction set to contain the true obstacle position.

Conformal prediction is an uncertainty quantification framework built on top the empirical statistics. Let the cumulative distribution function $F_X \triangleq \mathbb{P}(Y(X) \leq \overline{Y}_\epsilon) = p$. Let $Q : [0,1] \to \mathbb{R}$ be a Quantile function, the $(1-\epsilon)^{th}$ quantile returns the value $\overline{Y}_\epsilon$ such that $Y \leq \overline{Y}_\epsilon$ for all $X \in \Omega$ as

$$Q(1-\epsilon) = F_X^{-1}(1-\epsilon).$$

[107, Lemma 1] relates that the desired $\overline{Y}_\epsilon$ is equivalently the $(1-\epsilon)^{th}$ quantile of the empirical distribution formed by nonconformity scores $\{Y_0, \ldots, Y_{n-1}, \infty\}$ with an additive ∞. To calculate the $(1-\epsilon)^{th}$ quantile, let $\{Y_{(0)}, \ldots, Y_{(n)}\}$ be a non-decreasing sorting of $\{Y_0, \ldots, Y_{n-1}, \infty\}$ where $Y_{(i)} \leq Y_{(i+1)}, \forall i \in \mathbb{Z}_0^n$, also known as the the order statistics of $\{Y_i\}_{i=0}^n$. The integer $q_\epsilon \triangleq \lceil (n+1)(1-\epsilon) \rceil$ is the order index that corresponds to the $(1-\epsilon)$ percent confidence level empirically, where the multiplier $(n+1)$ is the finite sample adjustment. The $(1-\epsilon)^{th}$ quantile, $\overline{Y}_\epsilon \triangleq Y^{s,q_\epsilon}$ defines the desired $1-\epsilon$ prediction region satisfying the following probability statement:

$$\mathbb{P}[\|\tilde{f}(X)\| \geq \overline{Y}_\epsilon] \leq \epsilon \tag{2.10}$$

for any random variable $X \in Q$ and $\overline{Y}_\epsilon = Z_\epsilon^{\tilde{f}}$ as desired. The set

$$C_{\tilde{f}}(X) \triangleq \{\boldsymbol{y} \in \mathbb{R}^3 \mid \|\boldsymbol{y}\| \leq \overline{Y}_\epsilon\}$$

is commonly referred to as the $(1-\epsilon)$ confidence conformal prediction set.

For the tightness of the conformal probabilistic upper bound, [108] shows the conformal prediction set $C_{\tilde{f}}$, given $\epsilon \in (0,1)$, achieves the smallest average set of all possible prediction schemes C that offer the desired coverage guarantee if the sample nonconformity scores $\{Y_0, \cdots, Y_{N-1}\}$ reflects the true conditional probability:

$$\min_{C \in C} \mathbb{E}[C(X)], \quad \text{subject to (2.10).} \tag{2.11}$$

Adaptive Conformal Prediction [14], [109], [110]

In the domain of safety-critical motion planning and control, accurately quantifying the uncertainty of the prediction of an obstacle's future behavior is of paramount importance. Conformal prediction methods, however, cannot rigorously make uncertainty quantifications for time-series predictions due to their inherent assumption of exchangeability in the nonconformity scores $\{Y_0, \ldots, Y_{n-1}, \infty\}$. Because elements of the later portion of a time series signal may depend upon the specific values of preceding time series members, the temporal ordering of the time series is not exchangeable. The exchangeability assumption similarly does not hold when the nonconformity scores $\{Y_t^k\}_{k=1}^n$ represents prediction errors for time $t + k$ made at time t where $n \in \mathbb{Z}_{>0}$. Addressing this limitation, adaptive conformal prediction has been introduced and explored in recent studies [109]–[112] for its applicability to time series and stochastic scenarios with distribution shifts.

The core concept of adaptive conformal prediction is to adaptively adjust the prediction region $Y_{t+1} \leq \rho_{t+1}$ so that $\mathbb{P}(Y_{t+1} \leq \rho_{t+1}) \geq 1 - \epsilon$ is true for each time t. This requirement is achieved by setting $\rho_{t+1} \triangleq \bar{Y}_{(q_{t+1})}$, where q_{t+1} is an adaptive index computed based on the most recently acquired data.

Define a time varying desired $(1 - \epsilon)$ confidence prediction region $\rho_t \triangleq \bar{Y}_{q_{\epsilon,t}}$ where $q_{\epsilon,t} \triangleq \lceil (t+1)(1 - \epsilon_{t+1}) \rceil$ where the realized risk-level at time t denoted by variable ϵ_t, is adapted online. In this way, the prediction region ρ_t becomes a tuneable parameter by the choice of ϵ_t. To adaptively obtain the parameter ϵ_t, ideas from online learning are used, as

$$\epsilon_{t+1} \triangleq \epsilon_t + \gamma(\epsilon - \mathbf{1}_{\hat{Y}_t > \rho_t}) \tag{2.12}$$

where $\hat{Y}_t$ denotes the observed realization of Y_t at time t and where γ is a learning rate and $\epsilon \in [0, 1]$ is the desired risk level. The idea is that ϵ_{t+1} adapts to changes in the distribution of $\{Y_0, Y_1, \ldots, Y_t\}$ over time by using the information on how much the prediction region ρ_t overcovered ($\hat{Y}_t \ll \rho_t$) or undercovered ($\hat{Y}_t \gg \rho_t$) in the past. In terms of collision risks for the moving obstacle avoidance problem, the overcovered cases mean the prediction model is becoming more accurate thus the prediction region along the predicted values should contract to reflect. Conversely, if the prediction region is undercovered, the confidence of the unsafe set being correct should decay as well as an increase to the prediction region over time.

In [111], the authors present a fully adaptive conformal prediction (FACP) algorithm where a set of learning rates $\{\gamma_k\}_{k=1}^{n_\gamma}$ is used in parallel from which the best γ is selected adaptively. The FACP algorithm takes a candidate set of learning rates $\{\gamma_0, \gamma_1, \cdots\}$ and runs multiple versions of ACI in parallel with the candidate γ_k to obtain the optimal learning rate γ^* that produces the best conformal prediction sets.

Based on past performance (using a reweighting scheme that evaluates which $\{\gamma_k\}_{k=1}^{n_\gamma}$ provided the best coverage), the authors maintain a belief p_t^k at each time step t for each $\{\epsilon_t^k\}_{k=1}^{n_\gamma}$. The new update law for realized risk-level ϵ_t^k for learning rate γ_k is

$$\epsilon_{t+1}^k \triangleq \epsilon_t^k + \gamma_k(\epsilon - \mathbf{1}_{y_t > \rho_t^k})$$

where $\rho_t^k \triangleq \bar{Y}_{(q_t^k)}$ with a time varying conformal index $q_t^k \triangleq \lceil (t+1)(1-\epsilon_t^k) \rceil$. The conformal prediction set size at time t becomes $\rho_t \triangleq \bar{Y}_{(q_t)}$ with a weighted conformal index using the beliefs as

$$q_t \triangleq \lceil (t+1)(1 - \sum_{k=1}^{n_\gamma} p_t^k \epsilon_t^k) \rceil.$$

2.4 Finite-time Optimal Control Problem

Given control input $\boldsymbol{u} \in \mathcal{U} \subset \mathbb{R}^{n_u}$, this thesis studies the following continuous-time controlled nonlinear dynamical system

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}), \tag{2.13}$$

given states $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^{n_x}$. The function $f : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^{n_x}$ is a Lipschitz continuous function. Similarly, the governing dynamics for the discrete-time system is

$$\boldsymbol{x}_{k+1} = f^d(\boldsymbol{x}_k, \boldsymbol{u}_k), \tag{2.14}$$

where the function $f^d : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^{n_x}$ is the discretized nonlinear dynamics of f given time discretization $\Delta t = t_k - t_{k-1}$ and the input $\boldsymbol{u}_k$ is assume to be at zero-order hold from t_k to t_{k+1}. Let the positive integer n_h be the planning horizon.

Considering the following finite-time optimal control problem (FTOCP) for discrete-time systems given current time t and current state $\boldsymbol{x}_t$,

$$\{\boldsymbol{u}_k^*\}_{k=0}^{n_h-1} = \min_{\{\boldsymbol{u}_k\}_{k=0}^{n_h-1}} \sum_{k=0}^{n_h-1} \mathcal{L}_k(\boldsymbol{x}_k, \boldsymbol{u}_k) + \mathcal{L}_T(\boldsymbol{x}_{n_h}) \tag{2.15}$$

$$\text{s.t.} \quad \boldsymbol{x}_{k+1} = f^d(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{2.16}$$

$$f_{eq}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \mathbf{0} \tag{2.17}$$

$$f_{ineq}(\boldsymbol{x}_k, \boldsymbol{u}_k) \leq \mathbf{0} \tag{2.18}$$

$$\boldsymbol{x}_{k+1} \in \mathcal{X} \tag{2.19}$$

$$\boldsymbol{u}_k \in \boldsymbol{U} \quad \forall k \in \mathbb{Z}_0^{n_h-1} \tag{2.20}$$

$$\boldsymbol{x}_0 = \boldsymbol{x}_t \tag{2.21}$$

$$\boldsymbol{x}_{n^h} \in \mathcal{X}_T, \tag{2.22}$$

where the convex function $\mathcal{L}_k : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$ is the stage cost (cost-to-go) and convex function $\mathcal{L}_T : \mathcal{X} \rightarrow \mathbb{R}$ is the terminal cost. The equality constraint (2.16) describes the controlled system dynamics. Constraints (2.42) and (2.18) represent general equality and inequality constraints of the system states and inputs, respectively. Constraints (2.19), (2.20), (2.21), and (2.22) are state, input, initial state, and terminal state constraints. The solution to the above FTOCP, input sequences $\{\boldsymbol{u}_k^*\}_{k=0}^{n_h-1}$, is the desired control inputs given current state $\boldsymbol{x}_t$ that minimizes the control objective, such as reference tracking. Being a nonlinear, nonconvex optimization program, the solution's optimality and its computational tractability heavily depend on the initial guess used by the iterative solving techniques and the problem dimension. For practical real-time applications, a convex reformulation or a sampling-based reformulation of the nonlinear FTOCP is commonly employed to acquire a locally optimal solution in fixed controlling intervals [30], [90]–[92].

Sequential Convex Programming [90], [91]

Sequential convex programming (SCP) is a local optimization method for nonconvex optimization problems. As an overview, the method seeks to reformulate nonconvex constraints and costs by locally accurate convex approximations. Specifically, the nonconvex FTOCP (2.40)-(2.22) is an example of finding sub-optimal solutions.

For a general nonlinear discrete-time system with dynamics (2.14), linearization around its nominal trajectory $\{\boldsymbol{x}_{t+k}, \boldsymbol{u}_{t+k}\}_{k=0}^{n_h-1}$ can be applied. The trajectory $\{\boldsymbol{x}_{t+k}, \boldsymbol{u}_{t+k}\}_{k=0}^{n_h-1}$ is termed the *warm start trajectory*. The general linearized dynamics can take the following form

$$\boldsymbol{x}_{t+1} = A_t\boldsymbol{x}_t + B_t\boldsymbol{u}_t + \boldsymbol{c}_t, \tag{2.23}$$

where the matrices A_t, B_t, and $\boldsymbol{c}_t$ can be obtained from finding the local dynamics Jacobian given $\{\boldsymbol{x}_{t+k}, \boldsymbol{u}_{t+k}\}_{k=0}^{n_h-1}$ as:

$$A_t \triangleq I_{n_x} + \Delta T \frac{\partial f}{\partial \boldsymbol{x}_t}\bigg|_{(\overline{\boldsymbol{x}}_t, \overline{\boldsymbol{u}}_t)}, B_t \triangleq \Delta T \frac{\partial f}{\partial \boldsymbol{u}_t}\bigg|_{(\overline{\boldsymbol{x}}_t, \overline{\boldsymbol{u}}_t)}, \tag{2.24}$$

$$\boldsymbol{c}_t \triangleq f(\overline{\boldsymbol{x}}_t, \overline{\boldsymbol{u}}_t) - A_t\overline{\boldsymbol{x}}_t - B_t\overline{\boldsymbol{u}}_t. \tag{2.25}$$

The local Jacobian reformulation (2.24) can also be applied to general (possibly-)nonlinear inequality and equality constraint maps f_{eq} and f_{ineq} in (2.42) and (2.18) to acquire their local affine representation, respectively. Such affine (first-order Taylor expansion) approximation of the original constraint is only valid in a region close to the warm start reference. Trust region methods can be implemented to confine the convexified optimization program to search only in a reducing “ball" near the warm start and also ensure that SCP algorithms converge to a solution. Typically, the trust region at time integration w is chosen as

$$\mathcal{T}^{(w)} = \{\boldsymbol{x} \in \mathbb{R}^{n_x} \mid \|\boldsymbol{x} - \overline{\boldsymbol{x}}^{(w-1)}\| \leq \tau_0\tau_1^w\} \tag{2.26}$$

where $\tau_0 \in \mathbb{R}_+$ is the initial trust region and $\tau_1 \in (0, 1)$ is the trust reduction multiplier that dictates the shrinkage and refinement of the optimization feasible set. $\overline{\boldsymbol{x}}^{(w-1)}$ is the warm start trajectory which is conveniently selected to be the $(w-1)^{th}$ iteration of the SCP solution.

Model Predictive Path Integral [30]

Model Predictive Path Integral (MPPI) is a sampling-based method, leveraging a duality condition from information theory, to obtain local optimal controllers given potentially nonlinear cost and constraints. Consider input sequence $\{\boldsymbol{u}_i\}_{i=0}^t$ where $\boldsymbol{u} \in \mathbb{R}^{n_{\boldsymbol{u}}}$, define $\boldsymbol{v}_i \triangleq \boldsymbol{u}_i + \boldsymbol{\delta}_i$, $\forall i \in \{0, \ldots, t\}$ as the sequence of *perturbed inputs* where $\boldsymbol{\delta}_i$ are zero mean Gaussian input perturbations $\{\boldsymbol{\delta}_i\}_{i=0}^{t-1} \in \mathcal{N}(0, \Sigma_{\boldsymbol{u}})$. Denote the input trajectories to the algorithm as the following sequences

$$V_{t-1} \triangleq [\boldsymbol{v}_0, \boldsymbol{v}_0, \cdots, \boldsymbol{v}_{t-1}] \in \mathbb{R}^{n_{\boldsymbol{u}} \times t}, \tag{2.27}$$

$$\Delta_{t-1} \triangleq [\boldsymbol{\delta}_0, \boldsymbol{\delta}_1, \cdots, \boldsymbol{\delta}_{t-1}] \in \mathbb{R}^{n_{\boldsymbol{u}} \times t}, \tag{2.28}$$

$$U_{t-1} \triangleq [\boldsymbol{u}_0, \boldsymbol{u}_1, \cdots, \boldsymbol{u}_{t-1}] \in \mathbb{R}^{n_{\boldsymbol{u}} \times t}. \tag{2.29}$$

Following the result [113], the probability density function for V_t, denoted as $d_{V|U,\Delta}$, is

$$d_{V|U,\Delta} = \left((2\pi)^{n_{\boldsymbol{u}}} |\Sigma_{\boldsymbol{u}}|\right)^{\frac{-t}{2}} \exp\left(-\frac{1}{2}\sum_{i=1}^{t} \boldsymbol{\delta}_i^T \Sigma_{\boldsymbol{u}}^{-1} \boldsymbol{\delta}_i\right), \tag{2.30}$$

with corresponding probability distribution $\Omega_{U,\Sigma}$. Similarly, the probability distribution corresponding $U_t = 0$ denotes as $\Omega_{0,\Sigma}$, commonly referred to as the base distribution. Consider the stochastic trajectory optimization problem [113]:

$$U_{t-1}^* = \min_{\substack{\boldsymbol{u}_i \in \mathcal{U} \\ \forall i \in \mathbb{Z}_1^t}} \mathbb{E}_{\Omega(U,\Sigma)}\left[\mathcal{L}_T(\boldsymbol{x}_t) + \sum_{i=0}^{t-1} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{u}_i)\right], \tag{2.31}$$

where $\mathcal{L}_T$ and $\mathcal{L}$, respectively, denote terminal and stage costs.

Let $\phi_{\boldsymbol{x}_0}(V_{t-1})$ be the flow of system (4.1) generated by applying input sequence $\{\boldsymbol{v}_i\}_{i=0}^{t-1}$ from initial condition $\boldsymbol{x}_0$.

From an information theoretic perspective, optimization (2.31) can be converted into a probability-matching problem. Define the *Free Energy* $\mathcal{F}_{\mathcal{E}}$, a metric for performance as

$$\mathcal{F}_{\mathcal{E}} = \log\left(\mathbb{E}_{\Omega_{0,\Sigma}}\left[\exp\left(-\frac{S(V_t, \boldsymbol{x}_0)}{\lambda}\right)\right]\right),$$

where $\lambda > 0$ is commonly referred to as the *inverse temperature*, and the function $S(V_t, \boldsymbol{x}_0)$ results the cost of the trajectory generated using input sequence V_t from initial state $\boldsymbol{x}_0$.

Suppose Ω_F and Ω_D are two probability distributions that are *absolutely continuous* with each other which means that if the probability density for probability distribution Ω_F is zero, the probability distribution Ω_D is also zero [30]. The KL-Divergence, denoted as $\mathbb{KL}$, between Ω_F and Ω_D are defined as

$$\mathbb{KL}(\Omega_F \| \Omega_D) \triangleq \mathbb{E}_{\Omega_F}\left[\log\left(\frac{d_f(V)}{d_d(V)}\right)\right], \tag{2.32}$$

where d_f and d_d are the probability density function corresponding to distributions Ω_F and Ω_D, respectively. Using the KL-Divergence properties between two probability distributions that are *absolutely continuous* with each other, [113] showed that a lower bound to the optimal control problem (2.31) can be obtained using the free energy:

$$-\lambda\mathcal{F}_{\mathcal{E}} \leq \mathbb{E}_{\Omega_{U,\Sigma}}\left[\mathcal{L}_T(\boldsymbol{x}_t) + \sum_{i=0}^{t-1} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{u}_i)\right]. \tag{2.33}$$

Further, Williams *et. al.* [113] showed that a lower bound to the optimal control problem (2.31) can be obtained using free energy as:

$$-\lambda\mathcal{F}_{\mathcal{E}} \leq \mathbb{E}_{\Omega_{U,\Sigma}}\left[\mathcal{L}_T(\boldsymbol{x}_t) + \sum_{i=0}^{t-1}\mathcal{L}(\boldsymbol{x}_i,\boldsymbol{u}_i)\right]. \tag{2.34}$$

In other words, the negative free energy is a lower bound to the finite-time trajectory optimization problem (2.31).

Summarizing the results [30], [114], the following iterative update law, building upon importance sampling,

$$U_t^* = U_t + \sum_{j=1}^{N_{sample}} w(\Delta_t^j)\Delta_t^j, \tag{2.35}$$

$$w(\Delta_t^j) = \frac{1}{\eta}\exp\left(-\frac{C_{\boldsymbol{x}}}{\lambda} - \sum_{i=0}^{t-1}\boldsymbol{u}_i^T\Sigma_{\boldsymbol{u}}^{-1}(\boldsymbol{u}_i + 2\boldsymbol{\delta}_i^j)\right), \tag{2.36}$$

produces a locally optimal solution to the problem (2.31). The weighting term $w \in \mathbb{R}$ characterizes the "importance" of each sampled input perturbation $\{\Delta_t^j\}_{j=1}^{N_{sample}}$, where N_{sample} is the number of sampled input perturbations. The parameter η can be approximated as

$$\eta \approx \sum_{i=1}^{N_{sample}} \exp\left(-\frac{C_{\boldsymbol{x}}}{\lambda} - \sum_{i=0}^{t-1}\boldsymbol{u}_i^T\Sigma_{\boldsymbol{u}}^{-1}(\boldsymbol{u}_i + 2\boldsymbol{\delta}_i^j)\right)$$

using Monte-Carlo estimation [30].

In general, MPPI is a sampling-based alternative to optimization-based MPC algorithms, flexible to general cost functions and system dynamics. The aggregation law (2.35)-(2.36) is highly parallelizable for online receding-horizon planning, as demonstrated in UGV racing applications [30].

Warm Start

Both MPC-SCP and MPPI algorithm requires an initial guess of the solution $\{\boldsymbol{u}_i\}_{i=0}^{n_h-1}$. Selecting an initial guess that is sufficiently close to the true optimal solution is essential for the algorithm to converge fast and reliably [115]. It is well known that the receding horizon nature of MPC and MPPI can be exploited to obtain excellent initial guesses. At a time instant i, this can be achieved by *shifting* the optimization program solution from the previous timestep $i-1$ and by updating the guess of the final control input. Under certain conditions, a locally stable controller enforcing state and actuation constraints can be designed allowing the feasibility of the initial guess to be guaranteed [116]. Typically, simpler approaches are taken. By duplicating the final control input, *i.e.*, $\boldsymbol{u}_{n_h-1} = \overline{\boldsymbol{u}}_{n_h-2}$, a final state can be computed using the dynamics model and the shifted input sequence $\{\overline{\boldsymbol{u}}_0, \overline{\boldsymbol{u}}_1, \cdots, \overline{\boldsymbol{u}}_{n_h-2}, \overline{\boldsymbol{u}}_{n_h-2}\}$ from initial state $\boldsymbol{x}_0 = \boldsymbol{x}_t$. If the previous solution $\overline{\boldsymbol{u}}_{i-1}$ is a feasible solution to the optimization problem, the shifted input sequence will also be feasible for all but the last time step.

If the system is differentially flat, such as a differential-drive wheeled ground vehicle, a multirotor aerial system, or an aerial manipulator, the input sequence given the desired reference $\boldsymbol{p}^d(t)$ can be readily found using the flat outputs. More details can be found in [117]. The differential-driven ground vehicle example has the following kinematic model

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} v \\ \omega \end{bmatrix} = g(\boldsymbol{x})\boldsymbol{u} = f(\boldsymbol{x}, \boldsymbol{u}), \tag{2.37}$$

where the system state $\boldsymbol{x} = [x, y, \theta]^T \in \mathcal{X} \subset \mathbb{R}^3$ is comprised of the robot's inertial x, y position and its heading angle θ. The control inputs $\boldsymbol{u} = [v, \omega]^T \in \mathcal{U} \subset \mathbb{R}^2$ are the vehicle's linear and angular velocity with respect to the body frame x-axis and z-axis, respectively. Denote the differentially flat input sequence given a position reference $\boldsymbol{p}^d(t)$ as the desired flatness-based input reference $\boldsymbol{u}_d = [v_d, \omega_d]$, which can be computed as

$$v_d = \begin{cases} \frac{\dot{x}_d}{\cos(\theta_d)}, & \text{if} \sin(\theta_d) = 0 \\ \frac{\dot{y}_d}{\sin(\theta_d)}, & \text{otherwise} \end{cases}, \quad \omega_d = \begin{cases} \frac{a_d}{v_d}, & \text{if } v_d \neq 0 \\ 0 & \text{otherwise} \end{cases}, \tag{2.38}$$

where $a_d = \frac{-\sin(\theta_d)\ddot{x}_d + \cos(\theta_d)\ddot{y}_d}{v_d}$ and $\theta_d = \text{ATAN2}(\dot{y}_d, \dot{x}_d)$. The differentially flat state and input trajectories can be used as a warm-start if there are no additional equality and inequality constraints.

Robust MPC and MPPI [26]

As model-based methods, standard MPC and MPPI may exhibit suboptimal performance in the face of modeling uncertainties and disturbances. Tube-based [26], [31] and robust [32] model predictive planners have been proposed to address these challenges. Both of these methods integrate an ancillary controller, such as iterative Linear Quadratic Gaussian (iLQG), to improve tracking with measurement feedback and robustness to uncertainties. These methods rely on the use of sequential linearization and the assumption of additive white Gaussian noise limits their applicability to nonlinear systems with non-Gaussian model uncertainties. A brief review of key concepts and terminologies is provided in this section.

Consider a continuous-time nonlinear control affine system

$$\dot{\boldsymbol{x}} = \hat{f} = \underbrace{h(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}}_{f} + \underbrace{\boldsymbol{d}(t)}_{\tilde{f}} \tag{2.39}$$

where $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ is the system state, $\boldsymbol{u} \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ is the control input, and $\boldsymbol{d} \in \mathcal{D} \subset \mathbb{R}^{n_x}$ is an external disturbance.

The standard Robust MPC formulation involves a minimization-maximization (min-max) optimization to construct a policy $\pi : \mathcal{X} \times \mathbb{R} \to \mathcal{U}$ as follows [118]:

$$\pi_k = \{\boldsymbol{u}_k^*\}_{k=0}^{n_h-1} = \min_{\{\boldsymbol{u}_k\}_{k=0}^{n_h-1}} \max_{\{\boldsymbol{d}_k\}_{k=0}^{n_h-1}} \sum_{k=0}^{n_h-1} \mathcal{L}_k(\boldsymbol{x}_k, \boldsymbol{u}_k) + \mathcal{L}_T(\boldsymbol{x}_{n_h}) \tag{2.40}$$

$$\text{s.t.} \quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + (h_k^d(\boldsymbol{x}_k) + g_k^d(\boldsymbol{x}_k)\boldsymbol{u}_k + \boldsymbol{d}_k)\Delta T \tag{2.41}$$

$$\boldsymbol{x}_k \in \mathcal{X},\ \boldsymbol{u}_k \in \mathcal{U},\ \boldsymbol{d}_k \in \mathcal{D}, \tag{2.42}$$

where $\boldsymbol{d}_k$ is a discrete-time realization of the model discrepancy $\tilde{f}$ given nominal state $\boldsymbol{x}_k$ and input $\boldsymbol{u}_k$. Parameters $n_h \in \mathbb{Z}_{>0}$ and $\Delta t \in \mathbb{R}_{>0}$ are the finite-time planning horizon length and time discretization step size. The stage cost $\mathcal{L}_k$ can be a general function such as the squared distance between the vehicle position and the desired reference. The solution to (2.40)-(2.42) can be computationally intractable [26]. Nevertheless, the resulting robust FTOCP policy may lead to overly conservative control sequences.

An alternative solution, which still provides robustness to uncertainty, is to pair an *ancillary controller* κ for disturbance rejection, with the open-loop MPC control input as:

$$\pi = \boldsymbol{u}^* + \kappa(\boldsymbol{x}, \boldsymbol{x}^*), \tag{2.43}$$

where $\boldsymbol{u}^*(t)$ and $\boldsymbol{x}^*(t)$ are the optimal open-loop input and state trajectories computed using the nominal dynamics $\dot{x} = f$, respectively. The ancillary controller ensures the realized states $\hat{\boldsymbol{x}}$ of the uncertain system remain in a robust control invariant (RCI) tube around the nominal trajectory $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ from the initial state $\boldsymbol{x}_0$.

Definition 2. *Let $\mathcal{X}$ be the set of allowable state and define $\tilde{\boldsymbol{x}} \triangleq \boldsymbol{x} - \boldsymbol{x}^*$. The set $\Omega_{RCI} \subset \mathcal{X}$ is a* ***robust control invariant*** *(RCI) tube if there exist an ancillary controller $\kappa(\boldsymbol{x}, \boldsymbol{x}^*)$ such that if $\tilde{\boldsymbol{x}}(t_0) \in \Omega_{RCI}$, for all realizations of the disturbance and model uncertainties, $\tilde{\boldsymbol{x}} \in \Omega_{RCI}$, $\forall t \geq t_0$.*

To complete the Robust MPC, a constraint-tightened version of the nominal MPC problem (based on the model without the additive disturbance, *i.e.*, $\dot{\boldsymbol{x}} = f$ instead of (2.39)) can be solved to generate the open-loop trajectory pair $(\boldsymbol{x}^*_{RCI}, \boldsymbol{u}^*_{RCI})$.

Chapter 3

ONLINE UUA: DATA-DRIVEN MODEL UNCERTAINTY QUANTIFICATION

The models used for control synthesis are insightful, though oftentimes inaccurate and heavily reduced in dimension and complexity. Nevertheless, these models are the foundation for synthesizing controllers and planners for complex robotic systems even in unstructured environments, *e.g.*, quadrupeds, bipeds, drones, *etc* [1], [10], [119], [120]. Robustification to disturbances (e.g., to compensate for the gap between the reduced and full order models) is required for control and planning frameworks to function reliably on these complex systems [18], [121]–[124].

This chapter details a risk-aware model augmentation approach via learning disturbance models online that does not require *a priori* disturbance knowledge. This approach is sample-efficient as shown in Section 3.4, where less than a minute of flight data is required to make risk-aware control improvements on a drone mid-flight.

This chapter was adapted from: [125]

3.1 Preliminaries

Recent studies on the robust control of nonlinear systems center around input-to-state-safe control [123], [126], [127] and risk-aware control [11], [20], [21], [124], [128] among other techniques. These methods typically assume *a priori* knowledge of a model and possible disturbances (or at least the magnitude thereof) and employ control techniques designed to reject those known disturbances. On the other hand, learning-based approaches attempt to identify the underlying model [92], [129]–[134], in many cases through Gaussian Process Regression (GPR) [135].

However, assuming *a priori* knowledge of disturbances might not be accurate in real-world settings, and Gaussian process regression for model determination tends to be sample-complex and only uncover expected system behavior. While learning expected behavior is indeed useful, control predicated on expected models of system behavior might yield problematic behavior in safety-critical settings where risk-sensitive approaches are preferable [136], [137]. Skipping the model identification

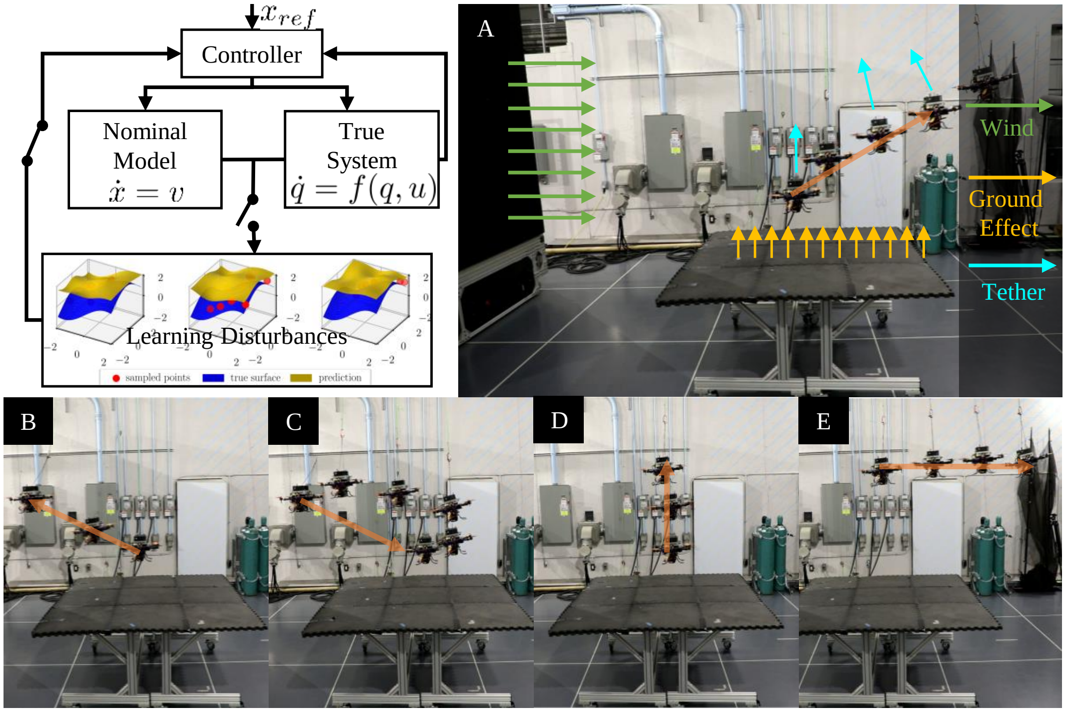

Figure 3.1: Online Unknown Uncertainty Quantification Overview. (Top Left) A general overview of the proposed procedure, (Top Right) a photo of the experimental setup, and (Bottom) snippets of flight paths taken by the drone during the second set of experiments run — the experiments depicted on the left in Figure 3.3. First, a nominal controller is implemented where the discrepancies between predicted model evolution and true system evolution are calculated. Then, Gaussian Process Regression is used to fit a risk-aware disturbance model for the disturbances that the nominal system experiences. Section 3.4 shows that the procedure dramatically improves baseline controller performance. Lastly, section 3.2 provides a statement on the theoretical accuracy of the learned model.

step, recent work in Bayesian Optimization and Reinforcement Learning aims to identify such risk-aware policies in a model-free fashion [22], [23], [138]–[141]. However, these prior works assume an ability to sample disturbances directly, assume *a priori* knowledge of disturbances, or are sample-complex.

A Brief Aside on Gaussian Process Regression (GPR)

A key concept developed in the chapter is the notion of *Surfaces-at-Risk* which is fitted via GPR as part of the procedure. GPR typically assumes the existence of an unknown function $f : X \to \mathbb{R}$ that is desirable to represent by taking noisy samples y of f at points $x \in X$ where the noise ξ is typically assumed to be sub-Gaussian [135], [142], [143]. Let $\mathcal{X} \triangleq \{x_i\}_{i=1}^N$ be a set of N scalars $x \in X$ and $\mathcal{Y} = \{y_i\}_{i=1}^N$ be the corresponding set of noisy observations, *i.e.*, $\{y_i = f(x_i) + \xi, \ \forall \ x_i \in \mathcal{X}\}$. Furthermore, let $k : X \times X \to \mathbb{R}$ be a positive-definite *kernel function*. Then, a *Gaussian Process* is uniquely defined by its mean function $\mu : X \longrightarrow \mathbb{R}$ and its variance function $\sigma : X \longrightarrow \mathbb{R}$. These functions are defined as follows, with $k_N(x) = [k(x, x_i)]_{x_i \in \mathcal{X}}$, $\mathbb{K} = [k(x_i, x_j)]_{x_i, x_j \in \mathcal{X}}$, $y_{1:N} = [y_i]_{y_i \in \mathcal{Y}}$, and $\lambda = (1 + \frac{2}{N})$:

$$\mu_N(x) = k_N(x)^T \left(\mathbb{K} + \lambda I_N\right)^{-1} y_{1:N}, \quad \sigma_N(x) = k_N(x, x), \tag{3.1}$$

$$k_N(x, x') = k(x, x') - k_N(x)^T \left(\mathbb{K} + \lambda I_N\right)^{-1} k_N(x'), \tag{3.2}$$

where I_N denotes an identity matrix of size N.

Lastly, each kernel function has a space of functions it can reproduce to point-wise accuracy, it is a *Reproducing Kernel Hilbert Space* (RKHS). Under the assumption that the function to-be-fitted f has bounded norm in the RKHS of the chosen kernel k, GPR guarantees high-probability representation of f as formalized in the theorem below [143]:

Theorem 1. *Let $f : X \longrightarrow \mathbb{R}$, $\mathcal{X} = \{x_i\}_{i=1}^N$ be a set of N points $x \in X$, $\mathcal{Y} = \{y_i = f(x_i) + \xi\}_{x_i \in \mathcal{X}}$ be a set of noisy observations y_i of $f(x_i)$ with R sub-gaussian noise ξ, and $k : X \times X \longrightarrow \mathbb{R}$ be a positive-definite kernel function. If f has B-bounded RKHS norm for some $B > 0$, i.e., $\|f\|_{RKHS} \leq B$, then, with μ_N and σ_N as per* (3.1) *and with minimum probability* $(1 - \epsilon)$*,*

$$|\mu_N(x) - f(x)| \leq \left(B + R \sqrt{2 \ln \frac{\sqrt{\det\left((1 + \frac{2}{N}) I_N + \mathbb{K}\right)}}{\epsilon}} \right) \sigma_N(x), \ \forall \, x \in X, \tag{3.3}$$

where operator $\det(\cdot)$ *is the matrix determinant.*

Surfaces-at-Risk for Scalar Stochastic Processes

This section formally defines a *Surface-at-Risk* for a scalar stochastic process — the specific structure fitted via GPR. Given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with Ω a sample space, $\mathcal{F}$ a σ-algebra over Ω defining events, and $\mathbb{P}$ a probability measure, defining a scalar *stochastic process* S over the indexed space $\mathcal{X}$ as a collection of scalar random variables $S_x : \Omega \longrightarrow \mathbb{R}$, *i.e.*, $S = \{S_x\}_{x \in \mathcal{X}}$. Here, each scalar random variable S_x has a (perhaps) different distribution $\pi_x : \mathbb{R} \longrightarrow [0, 1]$ such that probability of S_x taking values in $A \subseteq \mathbb{R}$, *i.e.*, $\mathbb{P}_{\pi_x}[S_x \in A \subseteq \mathbb{R}]$, is well-defined.

Risk measures are functions of these scalar random variables, and Value-at-Risk is a specific type of risk measure stemming from the financial literature [144].

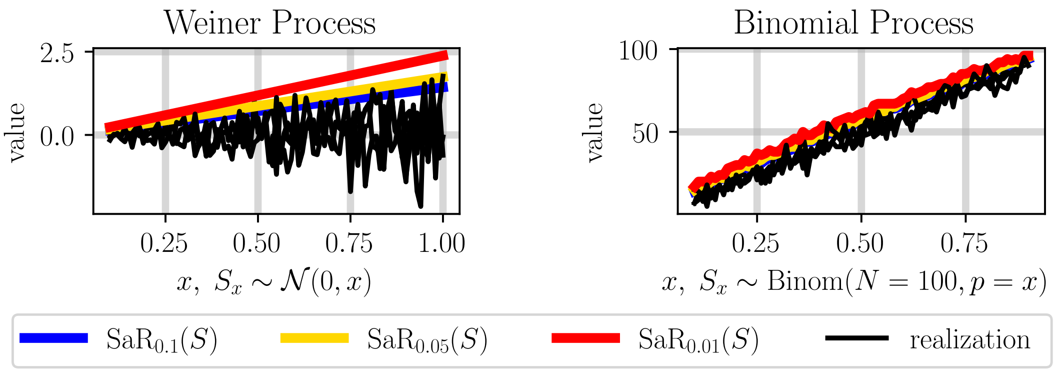

Figure 3.2: Example Surfaces-at-Risk at risk-levels $\epsilon \in [0.1, 0.05, 0.01]$ for a Weiner Process (Left) and Binomial Process (Right). Distributions for the indexed scalar random variables S_x comprising each process S are provided on the axes. Sample realizations of the stochastic processes are shown in black, with Surfaces-at-Risk shown via colored lines.

Definition 3. *The Value-at-Risk level $\epsilon \in [0, 1]$ of a scalar random variable X defined over the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with distribution p is defined as the $(1 - \epsilon)^{th}$ quantile of X:*

$$\text{VaR}_\epsilon(X) \triangleq c \quad \text{s. t.} \quad c = \inf\{z \in \mathbb{R} \mid \mathbb{P}_p[X \leq z] \geq 1 - \epsilon\}. \tag{3.4}$$

Then, the *Surface-at-Risk* for a scalar stochastic process is a similar collection of the Values-at-Risk of the underlying scalar random variables constituting the scalar stochastic process.

Definition 4. *The Surface-at-Risk level $\epsilon \in [0, 1]$ of a scalar stochastic process S indexed by the set $\mathcal{X}$ is the indexed collection of the Values-at-Risk level ϵ of each random variables S_x comprising S:*

$$\text{SaR}_\epsilon(S, x) = \text{VaR}_\epsilon(S_x). \tag{3.5}$$

Figure 3.2 shows a few examples of Surfaces-at-Risk for varying risk-levels ϵ overlaid on realizations of common (Weiner and Binomial) stochastic processes.

3.2 Learning Surface-at-Risk

The Risk-Aware Disturbance-Norm Identification Problem

From a risk-aware standpoint, the objective of this work is to identify a *Surface-at-Risk* as per Definition 4 for a scalar stochastic process S indexed over the model state-space $\mathcal{D}^{\boldsymbol{x}}$. Sample realizations of this process correspond to disturbance norms the system might experience at any given model state $\boldsymbol{x} \in \mathcal{D}^{\boldsymbol{x}}$. To formally state this problem, the true system state is denoted via $\hat{\boldsymbol{x}}$, and the sim (reduced-order) model state is denoted via $\boldsymbol{x}$, *i.e.*, $\forall\ k, j = 0, 1, 2, \ldots,$ (perhaps) different state and input spaces, and process noise $\boldsymbol{\xi}$ with (unknown and perhaps) state-dependent multivariate distribution Π

$$\begin{aligned} &\textbf{True:} \quad \hat{\boldsymbol{x}}_{t+1} = \hat{f} = f(\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{u}}_t) + \boldsymbol{\xi}_t, \quad \hat{\S}_t \in \hat{\mathcal{D}}^{\boldsymbol{x}},\ \hat{\boldsymbol{u}}_t \in \hat{\mathcal{D}}^{\boldsymbol{u}},\ \boldsymbol{\xi} \sim \Pi, \\ &\textbf{Sim:} \quad \boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t), \qquad\qquad\qquad \boldsymbol{x}_t \in \mathcal{D}^{\boldsymbol{x}},\ \boldsymbol{u}_t \in \mathcal{D}^{\boldsymbol{u}}. \end{aligned} \tag{SYS}$$

As an example consistent with the demonstration to follow, the true system would be a drone with a single integrator sim model. Specifically, the true state would be the drone's position and orientation, and the true input would be the rotor torques. Meanwhile, the model state would be the drone's position in 3-space, and the model input would be the desired velocity.

To identify the discrepancy between the systems in (SYS), two maps are defined - $P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}}$ which projects the true state $\hat{\boldsymbol{x}}$ to the model state $\boldsymbol{x}$ and $P_{\boldsymbol{u}\to\hat{\boldsymbol{u}}}$ which extends the model input $\boldsymbol{u}$ to the true input $\hat{\boldsymbol{u}}$, *e.g.*, $P_{\boldsymbol{u}\to\hat{\boldsymbol{u}}}$ provides rotor torques to realize the desired velocity in 3-space:

$$P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}} : \hat{\mathcal{D}}^{\boldsymbol{x}} \to \mathcal{D}^{\boldsymbol{x}}, \quad P_{\boldsymbol{u}\to\hat{\boldsymbol{u}}} : \mathcal{D}^{\boldsymbol{u}} \times \hat{\mathcal{D}}^{\boldsymbol{x}} \to \hat{\mathcal{D}}^{\boldsymbol{u}}. \tag{MAPS}$$

To note, the only assumption made is the existence of these maps and the ability to use them, there are no assumptions on the maps' uniqueness properties or the knowledge of the maps' analytic form, *etc*. To put these maps in the context of the drone example, the drone's underlying controller operates at 1 kHz making the true-system time step 1 ms. Since the typical model inputs are provided at 50 Hz frequency, $K = 20$. $P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}}$ is just the projection of the drone's position in 3-space, and $P_{\boldsymbol{u}\to\hat{\boldsymbol{u}}}$ is the on-board controller that takes in a commanded 3-space velocity — model input $\hat{u}$ — and updates rotor speeds at 1 kHz to achieve that velocity. These maps will be further explained in Section 3.4. Finally, an assumption is made that after some amount of true system time-steps $K > 0$, the observed projected true system evolution is observable. K denotes the *time-dilation parameter* and the observation function O is defined as follows:

$$\hat{\boldsymbol{x}}_{t+1} = \hat{f}(\hat{\boldsymbol{x}}_t, P_{\boldsymbol{u}\to\hat{\boldsymbol{u}}}(\boldsymbol{u}_t, \hat{\boldsymbol{x}}_t)), \quad O(\hat{\boldsymbol{x}}_0, \boldsymbol{u}) = P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}}(\hat{\boldsymbol{x}}_K). \tag{OBS}$$

Using these maps, the true system's projected evolution can be formally stated, *i.e.*, the evolution of $\boldsymbol{x}_j = P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}}(\hat{\boldsymbol{x}}_{Kj})$, when driven by a feedback controller $pi : \mathcal{D}^{\boldsymbol{x}} \to$

$\mathcal{D}^{\boldsymbol{u}}$. Comparing projected and sim model evolution results in the discrepancy d:

$$\begin{aligned}\boldsymbol{x}_{t+1} = f(P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}}(\hat{\boldsymbol{x}}_{Kt}), \pi(P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}}(\hat{\boldsymbol{x}}_{Kt})))+ \\ \underbrace{O(\hat{\boldsymbol{x}}_{Kt}, \pi(P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}}(\hat{\boldsymbol{x}}_{Kt})) - f(P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}}(\hat{\boldsymbol{x}}_{Kt}), \pi(P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}}(\hat{\boldsymbol{x}}_{Kt})))}_{\triangleq \boldsymbol{d},\text{ and } \delta=\|\boldsymbol{d}\| \text{ has distribution } \Delta_{\boldsymbol{x}}:\mathbb{R}\to[0,1]}.\end{aligned} \tag{3.6}$$

Then, inspired by the input-to-state-safe barrier and input-to-state-stable Lyapunov works whose robust controllers only require information on the 2-norm of this disturbance $\boldsymbol{d}$, it is desirable to learn a probabilistic upper bound on $\|\boldsymbol{d}\|$ by taking samples of indexed random variables $S_{\boldsymbol{x}}$ comprising a disturbance-norm stochastic process S indexed by $\mathcal{D}^{\boldsymbol{x}}$ as in (SYS).

Definition 5. *The disturbance-norm stochastic process $S = \{S_{\boldsymbol{x}}\}_{\boldsymbol{x}\in\mathcal{D}^{\boldsymbol{x}}}$ where samples of each random variable $S_{\boldsymbol{x}}$ correspond to norms δ of disturbances $\boldsymbol{d}$ as defined in equation* (3.6)*. The variability in norm samples δ arises through the assumed process noise $\boldsymbol{\xi}$ in the true system dynamics in* (SYS).

Remark on Residuals: If one only considers a deterministic discrepancy between the true and sim models, then the disturbances $\boldsymbol{d}$ as per (3.6) would correspond to residual dynamics, and the procedure would fit a surface to the norm of the residual dynamics (learning residual dynamics has a well-studied history, see [12], [145]–[147] and citations within). The discrepancy between these approaches and the proposed one is the learning of a probabilistic bound on the norm of any stochastic, model-state-dependent disturbances that affect the system during operation. This is the reason for representing the discrepancies as a stochastic process and fitting a Surface-at-Risk, which provides a natural way to reason about risk-aware disturbance rejection in a context including model errors and stochastic uncertainty.

Furthermore, the disturbance-norm stochastic process is assumed to be indexed over the model state space $\mathcal{D}^{\boldsymbol{x}}$ as opposed to the true state space $\hat{\mathcal{D}}^{\boldsymbol{x}}$. Moreover, this method assumes the ability to measure the projected state $\boldsymbol{x}_t = P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}}(\hat{\boldsymbol{x}}_{Kt})$. Therefore, sampled disturbance norms $\boldsymbol{\delta}$ can be matched to points in the projected state space $\mathcal{D}^{\boldsymbol{x}}$. Then, the goal is to identify a "close" upper bound to the *Surface-at-Risk* for this disturbance-norm stochastic process at some risk-level $\epsilon \in [0, 1]$.

Problem 1. *Identify an upper bound to the Surface-at-Risk at some risk-level $\epsilon \in [0, 1]$ for the disturbance-norm stochastic process S as per Definition 5 with Surfaces-at-Risk as defined in Definition 4. Specifically, identify an estimate $\mathbb{SR}_\epsilon$*

such that,

$$\mathbb{SR}_\epsilon(S, \boldsymbol{x}) \geq \mathrm{SaR}_\epsilon(S, \boldsymbol{x}), \; \forall\, \boldsymbol{x} \in \mathcal{D}^{\boldsymbol{x}}. \tag{3.7}$$

While the aforementioned upper bound $\mathbb{SR}_\epsilon$ could be arbitrarily large and satisfy (3.7), the objective is to find a "close" upper bound to the true Surface-at-Risk level ϵ to facilitate risk-aware control.

Fitting a Disturbance-Norm Surface-at-Risk

For identifying such an upper bound $\mathbb{SR}_\epsilon$, noting that even for stochastic processes whose sample realizations are non-differentiable, their Surfaces-at-Risk is relatively smooth — see Figure 3.2 for examples. Intuitively, the disturbance norms δ_i, δ_j at "close" model states $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{D}^{\boldsymbol{x}}$ is expected to be similarly "close":

Assumption 1. *For the disturbance-norm stochastic process S in Definition 5, the Surface-at-Risk at a given risk-level $\epsilon \in [0, 1]$ has bounded discrepancy. I.e., $\exists\, \alpha, \beta \in \mathbb{R}_{\geq 0}$ such that,*

$$\forall\, \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{D}^{\boldsymbol{x}}, \; \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \leq \alpha \implies |\, \mathrm{SaR}_\epsilon(S, \boldsymbol{x}_i) - \mathrm{SaR}_\epsilon(S, \boldsymbol{x}_j)| \leq \beta. \tag{3.8}$$

Notably, this assumption only implies a bounded discrepancy, and not continuity, *e.g.*, a bounded piecewise continuous function would have bounded variance as per the assumption. This assumption is verified to hold for the data set collected on hardware in Section 3.4.

Second, (perhaps noisy) unbiased samples of $\mathbb{SR}_\epsilon(S, \boldsymbol{x})$ for a given model state $\boldsymbol{x} \in \mathcal{D}^{\boldsymbol{x}}$ need to be considered. By equation (3.7), $\mathbb{SR}_\epsilon(S, \boldsymbol{x}) \geq \mathrm{VaR}_\epsilon(S_{\boldsymbol{x}})$. Define one sample δ_j of $S_{\boldsymbol{x}_j}$ as follows, where O is as per (OBS), and $P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}}(\hat{\boldsymbol{x}}_{Kt})$ is as per (MAPS):

$$\delta_t = \|O(\hat{\boldsymbol{x}}_{Kt}, \pi(P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}}(\hat{\boldsymbol{x}}_{Kt})) - f(P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}}(\hat{\boldsymbol{x}}_{Kt}), \pi(P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}}(\hat{\boldsymbol{x}}_{Kt}))\|,$$
$$\boldsymbol{x}_t = P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}}(\hat{\boldsymbol{x}}_{Kt}). \tag{3.9}$$

Then, multiple samples δ_j for sequential model states visited during operation are grouped, *i.e.*, $\delta_j, \delta_{j+1}, \ldots$ for $\boldsymbol{x}_j, \hat{x}_{j+1}, \ldots$ to produce an upper bound to at least one Value-at-Risk level ϵ of a sampled random variable, *i.e.*, $\mathrm{VaR}_\epsilon(S_{\hat{x}_j}), \mathrm{VaR}_\epsilon(S_{\hat{x}_{j+1}}), \ldots$. To do so, the following theorem is stated for N scalar random variables X with (perhaps) different distributions π.

```
Algorithm 1: Fitting a Disturbance-Norm Surface-at-Risk
Data: $\alpha, \beta$ for Assumption 1, an integer $\mathrm{N_{RV}} > 0$ for Proposition 1
      corresponding to the number of random variables to sample, time-step
      dilation parameter $K > 0$ between true system evolution and model
      evolution as per (OBS), and $k : \mathcal{D}^{x} \times \mathcal{D}^{x} \to \mathbb{R}$ a kernel function
Initialize: $\mathrm{s} = 0, \mathbb{X} = [], \mathbb{Y} = []$ ;
References: Disturbance Norm samples $\delta_j$ as per (3.9) and projector $P_{\hat{x} \to x}$ as
  per (MAPS) ;
while True do
    Initialize empty data-set, i.e., $\mathcal{S}_{\mathrm{s}} = [\ ]$ ;
    for $j = \mathrm{N_{RV}} \cdot \mathrm{s}, \mathrm{N_{RV}} \cdot \mathrm{s} + 1, \dots, \mathrm{N_{RV}}(\mathrm{s} + 1) - 1$ do
        Collect state-indexed disturbance norm samples, i.e.,
          $\mathcal{D}_{\mathrm{s}} \leftarrow \mathcal{D}_{\mathrm{s}} \cup (\delta_j, \boldsymbol{x}_j = P_{\hat{x} \to x}(\hat{\boldsymbol{x}}_{Kj}))$;
    end
    Augment GP state dataset with $\mathcal{S}_{\mathrm{s}}$: $\mathbb{X} \leftarrow \mathbb{X} \cup \boldsymbol{x}_{\mathrm{N_{RV}}(\mathrm{s}+1)-1}$;
    Augment GP norm dataset with $\mathcal{S}_{\mathrm{s}}$: $\mathbb{Y} \leftarrow \mathbb{Y} \cup \max\{\delta_\ell \in D\} + \beta$ ;
    Fit $\mu_{\mathrm{s}}, \sigma_{\mathrm{s}}$ as per (3.1) with data sets $\mathbb{X}, \mathbb{Y}$. $\mathrm{s}{+}{+}$ ;
end
```

Proposition 1. *Let $\{X_i\}_{i=1}^N$ be a collection of N scalar random variables with (perhaps) different distributions $\{\pi_i\}_{i=1}^N$, and let $\{x_i\}_{i=1}^N$ be a set of N samples of these random variables, one sample per each random variable, i.e., x_i is a sample of X_i. Then, for any $\epsilon \in [0,1]$, the probability that at least one sample $x_\ell \in \{x_i\}_{i=1}^N$ is greater than the Value-at-Risk level ϵ of its corresponding random variable X_ℓ is equivalent to $1 - (1-\epsilon)^N$, i.e., with* VaR *as per Definition 3 and* $\forall\ \epsilon \in [0,1]$,

$$\mathbb{P}_{\pi_1, \pi_2, \dots, \pi_N}\left[\exists\, x_\ell \in \{x_i\}_{i=1}^N \text{ s.t. } x_\ell \geq \mathrm{VaR}_\epsilon(X_\ell)\right] \geq 1 - (1-\epsilon)^N. \tag{3.10}$$

Proof. Consider a random variable $X_\ell \in \{X_i\}_{i=1}^N$. The probability of taking a sample x_ℓ of X_ℓ such that $x_\ell \geq \mathrm{VaR}_\epsilon(X_\ell)$ is less than or equal to ϵ by Definition 3. The same line of reasoning holds $\forall\ X_\ell \in \{X_i\}_{i=1}^N$. As such, the probability that no sample $x_\ell \in \{x_i\}_{i=1}^N$ is greater than the corresponding Value-at-Risk level ϵ is less than or equal to $(1-\epsilon)^N$, yielding the result. □

The procedure for generating unbiased samples of the upper bound $\mathbb{SR}_\epsilon$ stems directly from Proposition 1 and Assumption 1. Let the system evolve for $\mathrm{N_{RV}}$ model time-steps and collect one norm sample δ_j per model state $\hat{x}_j$ visited during operation. This norm sample δ_j is calculated as per (3.9). Second, Proposition 1 guarantees that the largest norm sample δ_j^* is greater than the Value-at-Risk level ϵ for its corresponding indexed random variable $S_{x_j^*}$ with some minimum probability.

Third, if all norm samples were drawn from indexed random variables $S_{\boldsymbol{x}_j}$ whose indices $\boldsymbol{x}_j$ were “close", *i.e.*, $\|\boldsymbol{x}_s - \boldsymbol{x}_r\| \leq \alpha \ \forall\ \boldsymbol{x}_r \neq \boldsymbol{x}_s \in \{\boldsymbol{x}_{j+i}\}_{i=0}^{N-1}$ and for some $\alpha > 0$, Assumption 1 can be leveraged to augment the largest norm sample δ_j^* by a constant $\beta > 0$. The sum is, with minimum probability $1 - (1-\epsilon)^N$, an unbiased, non-noisy sample of $\mathbb{SR}_\epsilon(S, \boldsymbol{x}_j)$. Algorithm 1 formalizes this procedure and the main theoretical result follows.

Theorem 2. *Let* $\alpha, \beta, \mathrm{N_{RV}}, \mathrm{s}, \mu_\mathrm{s}, \sigma_\mathrm{s}$, *and* k *be as defined in Algorithm 1, let* $B > 0$, *let* SaR *be the Surface-at-Risk measure as per Definition 4 for some risk-level* $\epsilon \in [0, 1]$, *let* S *be the disturbance-norm stochastic process as per Definition 5, and let Assumption 1 hold for each data set* $\mathcal{S}_\mathrm{s}$ *in lines 5-7 of Algorithm 1 with respect to the given parameters* α, β. *If* $\| \mathbb{SR}_\epsilon(S)\|_{RKHS} \leq B$, *then with minimum probability* $\left(1 - (1-\epsilon)^{\mathrm{N_{RV}}}\right)^\mathrm{s}$ *the following holds* $\forall\ \boldsymbol{x} \in \mathcal{D}^{\boldsymbol{x}}$ *and* $\forall\ \ \mathrm{s} = \mathbb{Z}_1^\infty$*:*

$$|\mu_\mathrm{s}(\boldsymbol{x}) - \mathbb{SR}_\epsilon(S, \boldsymbol{x})| \leq B\sigma_\mathrm{s}(\boldsymbol{x}), \qquad \mu_\mathrm{s}(\boldsymbol{x}) + B\sigma_\mathrm{s}(\boldsymbol{x}) \geq \mathrm{SaR}_\epsilon(S, \boldsymbol{x}). \tag{3.11}$$

Proof. First, by the assumptions above, for each data set $\mathcal{S}_\mathrm{s}$ in lines 5-7 of Algorithm 1, one sample δ_j of $\mathrm{N_{RV}}$ (potentially) different random variables $S_{\hat{x}_j}$ is taken. By Proposition 1 and with minimum probability $1-(1-\epsilon)^{\mathrm{N_{RV}}}$, the maximum sample $\delta_j^* \triangleq \max\{\delta_\ell \in D\}$ is greater than the Value-at-Risk of its corresponding random variable $\mathrm{VaR}_\epsilon(S_{\boldsymbol{x}_j^*})$ (VaR is defined in Definition 3). If assumption 1 holds for each such set of random variables, then with minimum probability $1 - (1-\epsilon)^{\mathrm{N_{RV}}}$, the sum δ_j^* +β is greater than the value-at-risk level ϵ of any sampled random variable, *i.e.*, the sum δ_j^* +β is a non-noisy estimate of $\mathbb{SR}_\epsilon(S, \boldsymbol{x}),\ \forall\, \boldsymbol{x} \in \mathcal{S}_\mathrm{s}$. Hence, repeating this same argument for each data point in $\mathbb{X}, \mathbb{Y}$ and setting $R = 0$, as each sampled point is a non-noisy sample of the upper-bounding surface, the results of Theorem 1 with minimum probability $(1 - (1-\epsilon)^{\mathrm{N_{RV}}})^\mathrm{s}$ is recovered:

$$|\mu_\mathrm{s}(\boldsymbol{x}) - \mathbb{SR}_\epsilon(S, \boldsymbol{x})| \leq B\sigma_\mathrm{s}(\boldsymbol{x}),\ \forall\, \boldsymbol{x} \in \mathcal{D}^{\boldsymbol{x}}. \tag{3.12}$$

The final result holds by unraveling the absolute-value inequality in (3.12), as $\mathbb{SR}_\epsilon(S)$ is an upper-bounding surface for $\mathrm{SaR}_\epsilon(S)$. □

3.3 Hardware Validation

Implementation Specifics

All flight tests are performed at the Caltech Center for Autonomous Systems and Technology arena which is equipped with an Optitrack motion capture system that samples and streams the rotor-craft pose at 190 Hz. A safeguard tether is attached to

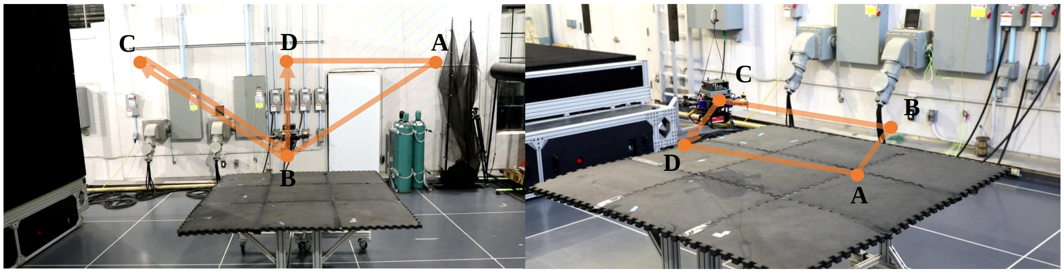

Figure 3.3: Depictions of the two types of periodic trajectories implemented in the drone experiments described in Section 3.4. These trajectories approximate difficult types of behaviors commonly asked of drones.

the drone (weights 2.46 kg) with a ~200 g passive weight attached on the other end to partially eliminate tether slack, which is another source of uncertainty. Figure 3.3 depicts the two types of flight paths taken, wherein the objective is to realize complex behaviors commonly asked of drones, *e.g.*, ascent and descent with both headwind and tailwind, circulating low to the ground, and taking off vertically in the presence of transverse wind. All disturbing winds were realized by The Caltech Real Weather Wind Tunnel, and windspeed information was not made available to the baseline controller to be augmented. This baseline controller was developed against a single integrator model, and as such, it outputs 3-space velocities at 50 Hz for the drone to follow. The velocities provided by this controller are tracked by the drone's onboard flight controller, a Hex Cube Orange running a PX4 autopilot [148].

With respect to the mathematical setting in Section 3.2, the true system dynamics are unknown though a single integrator model is employed as the sim model:

$$\begin{aligned} \boldsymbol{x}_{t+1} &= \boldsymbol{x}_t + \boldsymbol{u}_t(\Delta t = 0.02), \\ &\boldsymbol{x}_t \in \underbrace{[-2,2]^2 \times [1.2,2]}_{\mathcal{D}^{\boldsymbol{x}}}, \quad \boldsymbol{u}_j \in \underbrace{[-0.8,0.8]^2 \times [-0.5,0.5]}_{\mathcal{D}^{\boldsymbol{u}}}. \end{aligned} \tag{EXP-SYS}$$

The state projection map $P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}}$ as in (MAPS) reads the drone's position in 3-space. The input map M_u corresponds to the onboard PX4 controller that maps true drone states $\boldsymbol{x} \in \mathcal{D}^{\boldsymbol{x}}$ and commanded 3-space velocities $\boldsymbol{u} \in \mathcal{D}^{\boldsymbol{u}}$ to rotor speeds at 1 kHz. As the controller updates these desired velocities at 50 Hz, the time-dilation parameter $K = 20$ for Algorithm 1. Finally, the observation function O as per (OBS) outputs the projected true-system 3-space position after K true-system time-steps, and the disturbance-norm samples δ as per (3.9) are defined as follows:

$$\begin{aligned} \delta_t = \|O(\boldsymbol{x}_{Kt}, \pi(P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}}(\hat{\boldsymbol{x}}_{Kt}))) - (\boldsymbol{x}_j + \pi(P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}}(\hat{\boldsymbol{x}}_{Kt}))\Delta t)\|, \\ \boldsymbol{x}_j = P_{\hat{\boldsymbol{x}}\to\boldsymbol{x}}(\hat{\boldsymbol{x}}_{Kt}). \end{aligned} \tag{3.13}$$

The baseline controller $\pi : \mathcal{D}^x \rightarrow \mathcal{D}^u$ is a discrete-time Lyapunov controller designed to send the single-integrator system to a provided waypoint and does not take into account complex aerodynamic effects, *e.g.*, ground effects, transverse wind, and tethered disturbances, which are challenging to model and can degrade flight performance when ignored [92], [149]. Furthermore, the number of random variables sampled per data-collection step $\mathrm{N}_{\mathrm{RV}} = 60$ was kept constant. The squared-exponential kernel function with length-scale parameter $\ell = 1.0$ is used for all experiments.

The desired outcomes were twofold. First, an upper bound to the disturbance-norm *Surface-at-Risk* level $\epsilon = 0.05$ is acquired throughout one traversal of the desired flight path. In this initial flight path, only the baseline controller is implemented and augmented if the system takes longer than 10 seconds to reach within 0.1 m of the subsequent waypoint along the desired path. As each path comprises fewer than 6 waypoints, this ensures that the learned model considers less than a minute of data for all experiments on both flight paths. These cutoff times were specifically chosen to highlight the efficiency of the method with limited data. Second, on all subsequent flight paths, the fitted surface on the norm of disturbances that the Lyapunov controller should reject is provided while providing velocity commands. As such, performance improvements are expected using the augmented controller in the form of traversal time speedups through the series of waypoints, as subsequent waypoints are provided once the drone reaches within 0.1 m of the current, commanded waypoint, and the drone's controller should account for the vast majority of disturbances caused by wind, ground, and tether effects as an upper bound to the disturbance-norm *Surface-at-Risk* level $\epsilon = 0.05$ is fitted.

3.4 Results and Discussion

Four sets of experiments are conducted: (A) Hovering and moving while maintaining a 0.15 m height above ground (see right in Figure 3.3); (B) Ascent, descent, and vertical take-off without any wind (see left in Figure 3.3); (C) The same flight path as (B) but with a 0.6 m/s transverse wind. The wind flows from left to right when looking at the setup in Figure 3.3. A graphical example is also provided in Figure 3.1; (D) The same flight path as (B) and (C) but with a 2 m/s transverse wind.

Figure 3.4 shows the fitted $\mathrm{SaR}_{\epsilon=0.05}$ for each of the four experiments (A)-(D) ran on the drone, as labeled prior. As mentioned, in all cases there is at least a 2× speedup in flight path times when implementing the augmented controller, with as much as a

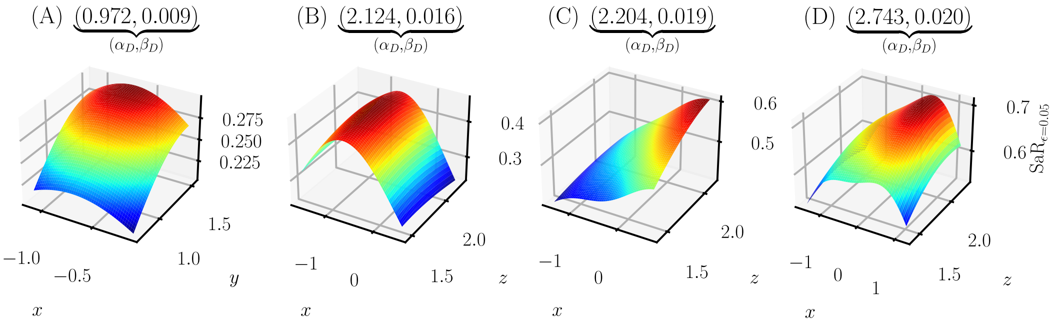

Figure 3.4: Fitted $\mathrm{SaR}_{\epsilon=0.05}$ for the four experiments depicted in Figure 3.3, with α_D the maximum distance between two sampled states for GPR, and β_D the maximum discrepancy between two sampled disturbance norms. Over all four experiments, there is a consistent 2× speedup in flight path times after implementation of the augmented controller — a qualitative result expected as per Theorem 2, as an upper bound is fitted to disturbance norms at 95% probability. This information is further explained in Section 3.4.

5× speedup in the hovering case (A). Furthermore, assumption 1 with respect to the data sets collected can be verified to hold for each experiment. Specifically, for (A), an assumption is made for states within $\alpha = 1$m their Values-at-Risk level $\epsilon = 0.05$ would not change by more than $\beta = 0.05$. As can be seen in the title of the associated subfigure in Figure 3.4, the reported values from data are smaller than their assumed counterparts, indicating that Assumption 1 held over this experiment, at least with respect to the collected data. For the remaining experiments, the assumed α, β values were as follows: (B) $\alpha = 3$m, $\beta = 0.05$; (C) $\alpha = 3$m, $\beta = 0.1$; (D) $\alpha = 3$m, $\beta = 0.2$. Therefore, as observed from the associated titles in Figure 3.4, assumption 1 can be similarly verified to hold over each of these cases as well — at least with respect to the data collected. As such, a significant increase in performance according to Theorem 2 is expected as was realized in all four cases with respect to flight path time speedups. All experiments can also be seen in the supplementary video here: [150].

Concluding Remarks

The results were threefold. A formal definition of the concept, Surfaces-at-Risk, is an extension of Value-at-Risk to scalar-valued stochastic processes. Second, the discrepancy between simulator and true-system evolution can be realized as a stochastic process, and a method to fit an upper bound to this process's *Surface-at-Risk* is provided. Third, a theoretical statement on the accuracy of the proposed approach with respect to fitting such an upper bound is carefully stated. Finally, the utility of the procedure is showcased in facilitating risk-aware control by implementing the procedure on a drone mid-flight and exhibiting dramatic performance improvements as a result.

Chapter 4

OFFLINE UUA: MATCHED AND UNMATCHED MODEL DISCREPANCIES

This chapter presents a novel data-driven multi-layered planning and control framework for the safe navigation of unmanned ground vehicles (UGVs) in the presence of unknown stationary obstacles and additive modeling uncertainties. The foundation of this framework is a novel robust model predictive planner, designed to generate optimal collision-free trajectories given an occupancy grid map, and a pairing ancillary controller, augmented to provide safety and robustness against model uncertainties extracted from data.

To tackle modeling discrepancies, both matched (input discrepancies) and unmatched (uncontrollable drift) model residuals between the true and the nominal reduced-order models using closed-loop tracking errors are identified from training data. Utilizing conformal prediction, probabilistic upper bounds for the model residuals are extracted which serve to augment the ancillary controller. Further, maximum tracking discrepancies are formulated, also known as the robust control invariance tube, under the augmented policy, which is used as collision buffers. Employing a LiDAR-based occupancy map to characterize the environment, a discrepancy-aware cost map is constructed, using these collision buffers, and integrated into a sampling-based model predictive path planner. The proposed framework in this chapter generates optimal and safe trajectories that can be tracked by the augmented ancillary controller robustly in the presence of model mismatches.

The effectiveness of the framework is experimentally validated in two applications: (1) autonomous high-speed trajectory tracking in a cluttered environment with four different vehicle-terrain configurations and (2) driver-assist application providing collision avoidance corrections based on user joystick commands.

This chapter was adapted from: [17]

4.1 Problem Statement

This chapter investigates the traversal of autonomous ground vehicles, encompassing differential drive robots, tracked vehicles, and skid steer vehicles. A widely used

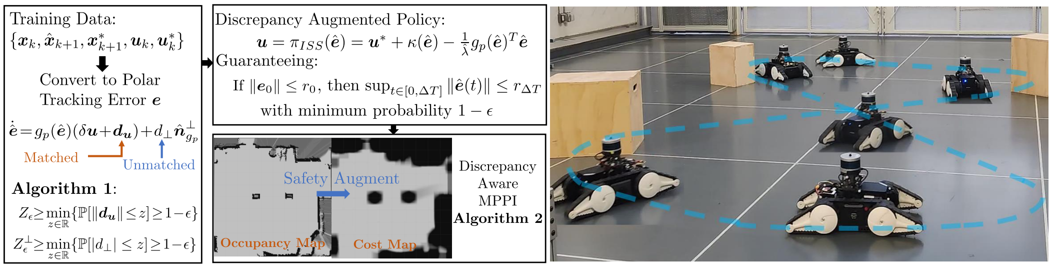
Training Data:
$\{\boldsymbol{x}_k, \hat{\boldsymbol{x}}_{k+1}, \boldsymbol{x}^*_{k+1}, \boldsymbol{u}_k, \boldsymbol{u}^*_k\}$
Convert to Polar Tracking Error $\boldsymbol{e}$
$\dot{\hat{\boldsymbol{e}}} = g_p(\hat{\boldsymbol{e}})(\delta\boldsymbol{u} + \boldsymbol{d}_{\boldsymbol{u}}) + d_\perp \hat{\boldsymbol{n}}^\perp_{g_p}$
Matched
Unmatched
Algorithm 1:
$Z_\epsilon \geq \min_{z\in\mathbb{R}}\{\mathbb{P}[\|\boldsymbol{d}_{\boldsymbol{u}}\| \leq z] \geq 1-\epsilon\}$
$Z^\perp_\epsilon \geq \min_{z\in\mathbb{R}}\{\mathbb{P}[|d_\perp| \leq z] \geq 1-\epsilon\}$
Discrepancy Augmented Policy:
$\boldsymbol{u} = \pi_{ISS}(\hat{\boldsymbol{e}}) = \boldsymbol{u}^* + \kappa(\hat{\boldsymbol{e}}) - \frac{1}{\lambda} g_p(\hat{\boldsymbol{e}})^T \hat{\boldsymbol{e}}$
Guaranteeing:
If $\|\boldsymbol{e}_0\| \leq r_0$, then $\sup_{t\in[0,\Delta T]} \|\hat{\boldsymbol{e}}(t)\| \leq r_{\Delta T}$
with minimum probability $1-\epsilon$
Safety Augment
Occupancy Map
Cost Map
Discrepancy Aware MPPI
Algorithm 2

Figure 4.1: Offline Unknown Uncertainty Quantification Overview. (Top) An overview of the three-step procedure of the proposed framework, summarizes the safety-critical framework: (1) offline data-driven model discrepancy identification and learning, (2) augmentation of the control policy based on the learned upper bounds and the associated collision buffers, and (3) a discrepancy-aware MPPI algorithm provides receding-horizon safe trajectory and input pairs. (Bottom) snapshots of the UGV movement in high-speed trajectory tracking in a cluttered environment, given unsafe desired trajectories.

model of such a vehicle is adopted for controller design and motion planning. This model, as seen in [151], [152], is a kinematic model described as:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} v \\ \omega \end{bmatrix} = g(\boldsymbol{x})\boldsymbol{u} = f(\boldsymbol{x}, \boldsymbol{u}), \tag{4.1}$$

where the system state $\boldsymbol{x} = [x, y, \theta]^T \in \mathcal{D}^{\boldsymbol{x}} \subseteq \mathbb{R}^3$ consists of the robot's inertial x, y position and its heading angle θ. The control inputs $\boldsymbol{u} = [v, \omega]^T \in \mathcal{D}^{\boldsymbol{u}} \subseteq \mathbb{R}^2$ are the vehicle's linear and angular velocity in the body x- and z-axes, respectively. The sets $\mathcal{D}^{\boldsymbol{x}}$ and $\mathcal{D}^{\boldsymbol{u}}$ denote the set of allowable states and inputs.

The nominal model (4.1) is a reduced-order description of a general differential-driven ground vehicle: it assumes that the wheels or tracks contact the ground at all times, ignoring phenomena such as slipping, skidding, motor dynamics, and communication delays. A more comprehensive model, accounting for these neglected factors, can be formulated as:

$$\dot{\boldsymbol{x}} = \hat{f}(X) = f(\boldsymbol{x}, \boldsymbol{u}) + \tilde{f}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{u}, \boldsymbol{d}(t)), \tag{4.2}$$

where the true dynamics $\hat{f} : Q \to \mathbb{R}^3$ maps state space Q to the vehicle's true linear and angular velocities, $f(\boldsymbol{x}, \boldsymbol{u})$ is the nominal model (4.1), and $\tilde{f} : Q \to \mathbb{R}^3$ is an *a priori* unknown additive disturbance, that models the discrepancies–it is possibly a function of state $\boldsymbol{x}$, input $\boldsymbol{u}$, process noise $\boldsymbol{d}(t)$, and hidden states $\boldsymbol{h}$. The variable $X \in Q$ in equation (4.2) can be interpreted as a vehicle operating state that includes inputs that correspond to the vehicle velocity $\dot{\boldsymbol{x}}$ via function $\hat{f}$. To note, projections from the vehicle operating state space Q to the nominal state space $\mathcal{D}^{\boldsymbol{x}}$ and nominal system input space $\mathcal{D}^{\boldsymbol{u}}$ are defined for analyzing the discrepancy between the not fully observable true dynamics (4.2) and the nominal model (4.1).

Definition 6. *Let $\boldsymbol{x}_X$ be the projection of sample X onto $\mathcal{D}^{\boldsymbol{x}}$, denoted as $P_{X\to\boldsymbol{x}}$, is defined as*

$$P_{X\to\boldsymbol{x}}(X) \triangleq \boldsymbol{x}_X \in \mathcal{D}^{\boldsymbol{x}}, \tag{4.3}$$

where $\mathcal{D}^{\boldsymbol{x}}$ is the nominal (reduced-order) admissible states and $X \in Q$ is a sampled vehicle true state.

Note, only the existence projections $P_{X\to\boldsymbol{x}}$ and $P_{X\to\boldsymbol{u}}$ is assumed. Most importantly, this work does not assume these projections are unique nor knowledge of their analytic forms, *etc*. Now, the true dynamics (4.2) given operating state X satisfies Now, the true dynamics (4.2) given operating state X satisfies

$$\dot{\boldsymbol{x}}_X = \hat{f}(X) = f(\boldsymbol{x}_X, \boldsymbol{u}_X) + \tilde{f}(X). \tag{4.4}$$

For notation simplicity, the subscripts X for $\boldsymbol{x}_X$ and $\boldsymbol{u}_X$ are omitted and referred to as the nominal state given any operating sample X.

Adhering to many robust optimal control works ([26]), the following type of additive model uncertainties, satisfying Assumption 2, is studied.

Assumption 2. *The true system dynamics $\hat{f}$ can be expressed as $\hat{f} = f + \tilde{f}$ where f is the nominal dynamics and $\tilde{f}$ represents all model uncertainties.*

The proposed approach is based on the notion that the vehicle aims to follow a desired (denoted by $(\cdot)^d$) reference path $\boldsymbol{p}^d(t) = [x^d(t), y^d(t)]^T$, which is designed without considering obstacles or model simplifications and mismatches. Since the nominal model (4.1) is differentially flat, the input sequence given the desired reference $\boldsymbol{p}^d(t) = [x^d(t), y^d(t)]^T$ can be readily found using (4.5), without considering obstacles. Denote the differentially flat input sequence given a position reference $\boldsymbol{p}^d(t)$ as the desired flatness-based input reference $\boldsymbol{u}^d = [v^d, \omega^d]$, which can be computed as

$$v^d = \begin{cases} \frac{\dot{x}^d}{\cos(\theta^d)}, & \text{if} \sin(\theta^d) = 0 \\ \frac{\dot{y}^d}{\sin(\theta^d)}, & \text{otherwise} \end{cases}, \quad \omega^d = \begin{cases} \frac{a^d}{v^d}, & \text{if } v^d \neq 0 \\ 0 & \text{otherwise} \end{cases}, \tag{4.5}$$

where $a^d = \frac{-\sin(\theta^d)\ddot{x}^d + \cos(\theta^d)\ddot{y}^d}{v^d}$ and $\theta^d = \text{ATAN2}(\dot{y}^d, \dot{x}^d))$.

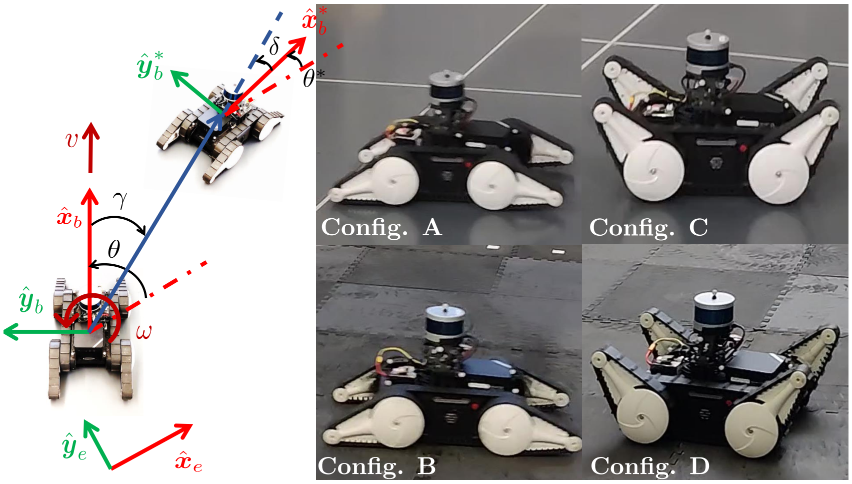

Figure 4.2: (Left) Definition of the polar coordinate. Subscripts $(\cdot)_b$, $(\cdot)_e$ denote the body and inertial frames, respectively. The pose with superscript $(\cdot)^*$ is the desired pose. (Right) Four experimentally validated scenarios. Config. A: UGV with flipper on rubber flooring. Config. B: UGV with flipper on foam flooring. Config. C: UGV without flipper on rubber flooring. Config. D: UGV without flipper foam flooring.

Ancillary Controller - Nominal Model

Since the open-loop inputs $\boldsymbol{u}_d(t)$ or $\boldsymbol{u}^*(t)$ can exhibit poor tracking when applied to the actual vehicle due to model simplifications and mismatches, the tracking error feedback is incorporated using [152] as the nominal ancillary controller,

$$\kappa(\boldsymbol{e}) = \delta\boldsymbol{u} \triangleq \begin{cases} v = k_1\rho\cos\gamma \\ \omega = k_2\gamma + k_1\frac{\sin(\gamma)\cos(\gamma)}{\gamma}(\gamma + k_3\delta) \end{cases}, \tag{4.6}$$

where k_1, k_2, and k_3 are positive constants. This ancillary controller drives the polar coordinates tracking error to zero with respect to the nominal model (4.1). The tracking error polar coordinates

$$\boldsymbol{e} = [\rho, \gamma, \delta]^T \triangleq f_{\boldsymbol{e}}(\boldsymbol{x}, \boldsymbol{x}^*)$$

is a function of the current vehicle state $\boldsymbol{x}$ and the optimal state $\boldsymbol{x}^*$, see Fig. 4.2.

Polar coordinate tracking error $\rho \triangleq \sqrt{(x - x^*)^2 + (y - y^*)^2}$ is the distance from the vehicle's current (x, y) position to a desired way point (x^*, y^*). Polar tracking error $\gamma \triangleq \tan^{-1}\frac{y-y^*}{x-x^*} - \theta$ denotes the angle between the vehicle's body-fixed x-axis (the longitudinal axis) and a vector that points from the vehicle's body-fixed frame origin to the desired waypoint (x^*, y^*). Similarly, $\delta = \gamma + \theta - \theta^*$ is the analogous angle between the vehicle's body-fixed x-axis to the optimal x-axis given by θ^*. The time derivative of the polar coordinates tracking error $\boldsymbol{e} \triangleq [\rho, \gamma, \delta]^T$ satisfies:

$$\dot{\boldsymbol{e}} = \begin{bmatrix} -\cos(\gamma) & \frac{\sin(\gamma)}{\rho} & \frac{\sin(\gamma)}{\rho} \\ 0 & -1 & 0 \end{bmatrix}^T \delta\boldsymbol{u} \triangleq g_p(\boldsymbol{e})\delta\boldsymbol{u}. \tag{4.7}$$

The vehicle operates in the domain $\mathcal{D}^{\boldsymbol{e}} = \mathcal{D}_\rho \times \mathcal{D}_\gamma \times \mathcal{D}_\delta \subset \mathbb{R}^3$ where $\mathcal{D}_\rho = (dz, \rho_{max})$, $\mathcal{D}_\gamma = (-\pi/2, \pi/2)$, and $\mathcal{D}_\delta = (-\pi/2, \pi/2)$. [1] Within domain $\mathcal{D}^{\boldsymbol{e}}$, the function $g_p(\boldsymbol{e})$ is bounded and Lipschitz continuous with Lipschitz constant l_{g_p}. The ancillary controller (4.6) is selected because of the following result given the operating domain $\mathcal{D}^{\boldsymbol{e}}$.

Lemma 1. *([152], [153]) Consider system* (4.7)*, control law* (4.6)*, and positive constants* k_1*,* k_2*, and* k_3*. The nominal closed-loop system is globally asymptotically stabilizing to* $\boldsymbol{e}^* = 0$.

Lemma 5 can be proved by the following valid and positive definite control Lyapunov function (CLF):

$$V(\boldsymbol{e}) = \frac{1}{2}(\rho^2 + \gamma^2 + k_3\delta^2). \tag{4.8}$$

Here are several properties of $V(\boldsymbol{e})$ which support the main theorem in Section 4.3. For quadratic form (4.8), there exist $\alpha_1, \alpha_2 \in \mathbb{R}_{>0}$ where $\alpha_1\|\boldsymbol{e}\|_2 \leq V \leq \alpha_2\|\boldsymbol{e}\|_2$, $\forall \boldsymbol{e} \in \mathcal{D}^{\boldsymbol{e}}$. Its time derivative

$$\dot{V} = -k_1\cos^2(\gamma)\rho^2 - k_2\gamma^2 = -\alpha_3(\|\boldsymbol{e}\|)$$

is negative semi-definite. Adhering to the Lyapunov framework, there exists a class $\mathcal{K}_\infty$ [154] function α_3 such that $\dot{V} \leq -\alpha_3(V)$. Parameters $\alpha_1, \alpha_2 \in \mathbb{R}_{>0}$, and function $\alpha_3(\cdot)$ play an important role in system safety and stability when model (4.7) is perturbed by unknown discrepancies, see Section 4.3.

The nominal trajectory tracking control policy

$$\boldsymbol{u} = \pi(\boldsymbol{e}) = \boldsymbol{u}^* + \kappa(\boldsymbol{e}) = \boldsymbol{u}^* + \delta\boldsymbol{u} \tag{4.9}$$

is employed which asymptotically tracks $\boldsymbol{x}^*$ without any model mismatches. The goal is to construct trajectories and control policies that are robust to model discrepancies and provide probabilistic guarantees to avoid *a priori* unknown obstacles.

[1] The vanishingly small interval $(0, dz)$ is a "deadzone" where the polar coordinate representation (4.7) is ill-conditioned as $dz \to 0$.

Modeling Discrepancies

A probabilistic framework is employed to describe model discrepancies and to define a probabilistic upper bound on the sum of all model discrepancies.

Definition 7. *Let function $\tilde{f} : Q \rightarrow \mathbb{R}^3$ measure the difference between the nominal model* (4.1) *and the true model* (4.2)*. Let $X \in Q$ denote a vehicle operating state where Q is the vehicle's true state space and $\tilde{f}(X) \in \mathbb{R}^3$ denote the additive disturbance at operating state X. Assuming the additive disturbance random variable has a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, a sample space Ω, a σ-algebra $\mathcal{F}$ over Ω defining events, and a probability measure $\mathbb{P}$. Define the* minimal model discrepancy upper bound*, $z_\epsilon^{\tilde{f}}$, given a risk level $\epsilon \in [0,1]$ over Ω for function $\tilde{f}$ as:*

$$z_\epsilon^{\tilde{f}} \triangleq arg \min_{z \in \mathbb{R}} \{\mathbb{P}[\|\tilde{f}(X)\| \leq z] \geq 1 - \epsilon\}, \quad \forall X \in Q. \tag{4.10}$$

The value $z_\epsilon^{\tilde{f}}$ can be interpreted as the $(1 - \epsilon)$ percentile of all possible model discrepancies norms, $\|\tilde{f}(X)\|$, for all $X \in Q$. Such differences, *i.e.* the **residual dynamics**, arise from too simplistic vehicle modeling. The nominal model (4.1) disregards some robot configurations (such as different flipper angles) and terrain types (rubber and foam in the experiments) as illustrated in Fig. 4.2.

The minimum value in (4.10) may only be reached given infinite samples of $\tilde{f}(X)$ or knowing the probability measure of X on Ω, which cannot be realized in practice. Instead, the objective is to identify positive values $Z_\epsilon^{\tilde{f}} \geq z_\epsilon^{\tilde{f}}$ given finite samples of $\boldsymbol{x}_X$, defined in (4.3), while making no assumptions on the distribution over Ω or the explicit form or uniqueness of the projection $P_{X \rightarrow \boldsymbol{x}}$. That is, $Z_\epsilon^{\tilde{f}}$ is a model discrepancy bound $\|\tilde{f}(X)\|$ for any $X \in Q$ given $\epsilon \in (0,1)$. It is not necessarily the tightest bound but a realizable bound from a limited data set.

Obstacle Avoidance

The safety-critical autonomy of ground vehicles traverses through incompletely known terrains populated with stationary but *a priori* unknown obstacles is discussed in this section. Assume that a 2-D occupancy grid map (O_t derived from sensor data) is available to the vehicle. The map, defined in inertial coordinates, has a width and length of $w_{map}, l_{map} \in \mathbb{Z}_{>0}$, respectively, and a grid resolution of r_{map}. Each grid cell is associated with an integer value between 0 and 100 that describes the cell occupancy probabilities where 0 indicates a 0% chance of obstacle occupancy and vice versa. Unobserved cells are assigned a risk-neutral 50% occupancy probability. The construction of the map is described in Section 4.4 which is built incrementally as the robot moves. This work is only interested in the dynamic uncertainties from model residuals and ignores possible sensing uncertainties. The proposed method can be extended to include measurement uncertainties.

Given this sensor-based grid map of the unknown environment, the proposed framework allows a ground vehicle with terrain modeling discrepancies to track a reference trajectory within the map while having probabilistic safety guarantees on collision avoidance. The criterion is:

$$\mathbb{P}[\boldsymbol{p}_t \notin O] > 1 - \epsilon \quad \forall t, \tag{4.11}$$

for any $\boldsymbol{p}_t \in \mathcal{V}_t$ where $\mathcal{V}_t$ denotes a uniform probability distribution over bounded set of x, y positions occupied the vehicle geometry at time t. The statement (4.11) is equivalent to collision-free with confidence $1 - \epsilon$ given occupied space O for all time.

Remark 1. *This chapter focuses on flat terrains populated with stationary obstacles. Nevertheless, the proposed framework can be readily extended to uneven surfaces using model* (4.1) *or a higher-fidelity model for rough terrain applications, such as the ones developed in [155], as the nominal model.*

Matched and Unmatched Discrepancies

Because the reduced-order model (4.1) is perturbed by an additive model discrepancy $\tilde{f}$, the discrepancy is decomposed into a matched component that lies in the range space of the actuation and an unmatched component that lies in the null space of the actuation matrix [156]–[158].

sw Let the range space of the differential-driven ground vehicle be

$$\text{match}(g) \triangleq \text{span}(g_1, g_2)(\boldsymbol{x}),$$

where g_1, g_2 are the columns of the actuation matrix $g(\boldsymbol{x})$. The null subspace for system (4.1) is similarly defined as

$$\text{umatch}(g) \triangleq \{\boldsymbol{x} \in \mathcal{D}^{\boldsymbol{x}} \mid g(\boldsymbol{x}) = \boldsymbol{0}\},$$

the null space of the actuation matrix.

Definition 8. *The additive model uncertainty $\tilde{f}$ can be decomposed into matched ($\tilde{f}_m$) and unmatched ($\tilde{f}_{um}$) discrepancies with respect to reduced-order model* (4.1). *Specifically, the matched discrepancy is the function $\tilde{f}_m : Q \to \mathbb{R}^3$ satisfying*

$$\tilde{f}_m(X) \in match(g(\boldsymbol{x}_X)), \quad \forall X \in Q. \tag{4.12}$$

The unmatched discrepancy is defined as the function $\tilde{f}_{um} : Q \to \mathbb{R}^3$ *satisfying*

$$\tilde{f}_{um}(X) \in umatch(g(\boldsymbol{x}_X)), \quad \forall X \in Q. \tag{4.13}$$

More importantly, for all samples $X \in Q$ *and their respective reduced-order state* $\boldsymbol{x}_X \in \mathcal{D}^{\boldsymbol{x}}$*, the summation* $\tilde{f}(X) = \tilde{f}_m(X) + \tilde{f}_{um}(X)$ *holds.*

4.2 Data-Driven Discrepancy Identification

This section summarizes the data-driven identification of a model discrepancy bound $Z_\epsilon^{\tilde{f}}$, where $\tilde{f}$ captures model mismatches in (4.2) from sources like slipping, skidding, change in surface traction properties, input delays, etc. Discrete-time data is captured from the continuous system at sampling interval ΔT. Time is indicated by an integer: variable $\boldsymbol{x}_i$ denotes the value of state $\boldsymbol{x}$ at time $t_i = i\Delta T$. N_T tuples of training data are gathered from running the system with the nominal tracking control law 4.9 and each tuple has the form

$$S_i = \{\boldsymbol{x}_{i-1}, \hat{\boldsymbol{x}}_i, \boldsymbol{x}_i^*, \boldsymbol{u}_{i-1}, \boldsymbol{u}_{i-1}^*\}, \tag{4.14}$$

where $\boldsymbol{x}_{i-1}$ is the state at time t_{i-1}, and the pair $(\boldsymbol{x}_i^*, \boldsymbol{u}_{i-1}^*)$ is the optimal state and the optimal input at t_i with respect to the cost in (2.31). This data set can be collected from multiple vehicle trajectories that result from an optimal trajectory planner (*e.g.*, by a model predictive control planner for dynamics (4.1)) of the form $(\boldsymbol{x}_i^*, \boldsymbol{u}_{i-1}^*)$ given current state $\boldsymbol{x}_{i-1}$.

The actual control input $\boldsymbol{u}_{t-1}$ applied to the vehicle in the interval $[t_{i-1}, t_i]$ is the sum of the optimal input $\boldsymbol{u}_{t-1}^*$ and and a correction term applied by the ancillary controller (4.6), $\delta\boldsymbol{u}_{t-1}$, given the polar coordinate tracking error at: $\boldsymbol{u}_{i-1} = \boldsymbol{u}_{i-1}^* + \delta\boldsymbol{u}_{i-1} = \boldsymbol{u}_{i-1}^* + \kappa(\boldsymbol{e}_{i-1})$. $\hat{x}_i$ is the system's actual state arising from the control input $\boldsymbol{u}_{i-1}$ when it is held constant during $t \in [t_{i-1}, t_i]$. The propagation of state $\boldsymbol{x}_{i-1}$ with input $\boldsymbol{u}_{i-1}$ through nominal model (4.1), the expected state $\boldsymbol{x}_{i-1}$ might be different from the measured $\hat{\boldsymbol{x}}_{i-1}$ because of unmodeled discrepancy $\tilde{f}$. Assuming that the optimal planner and ancillary controller are recomputed at each time step, and more specifically for the j tuples S_j, $\boldsymbol{x}_{j-1} = \hat{\boldsymbol{x}}_j$ holds. Figure 4.3 demonstrates the proposed identification process from training data as an overview.

Discrepancy Identification

To identify the matched and unmatched discrepancy, the polar coordinate tracking error $\boldsymbol{e}_{i-1}$ can be computed using $\boldsymbol{x}_{i-1}$ and $\boldsymbol{x}_i^*$ at time t_{i-1}. Then, the feedback (measured) tracking error at time t_i, denoted as $\hat{\boldsymbol{e}}_i$, is a function of $\hat{\boldsymbol{x}}_i$ and $\boldsymbol{x}_i^*$.

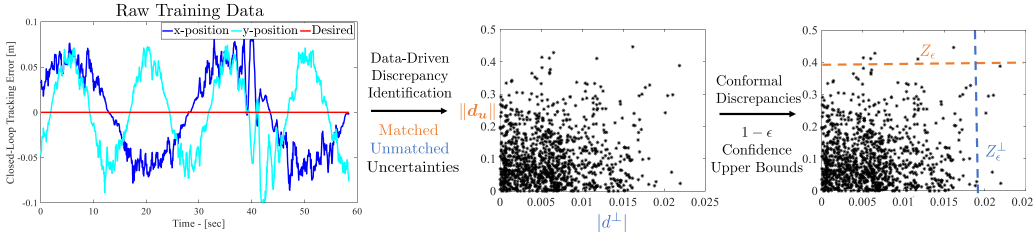

Figure 4.3: An outline of the proposed data-driven discrepancy identification method. From training data $\mathcal{S}_i$ (left panel), the closed-loop tracking errors are separated into matched and unmatched model discrepancies (center panel). Given a user-specified risk tolerance ϵ, conformal prediction is applied to obtain their probabilistic upper bounds (right).

Using (4.7), the polar coordinate tracking error can be calculated using the nominal error dynamics (4.7) at t_i given initial condition $\boldsymbol{e}_{i-1}$ and assuming $\delta\boldsymbol{u}_{i-1}$, generated by the ancillary controller, is the input as

$$\boldsymbol{e}_i = \int_{t_{i-1}}^{t_i} g_p(\boldsymbol{e}(\tau))\delta\boldsymbol{u}_{i-1} d\tau + \boldsymbol{e}_{i-1}, \tag{4.15}$$

where $\delta\boldsymbol{u}_{i-1}$ is zero-order hold during the interval $[t_{i-1}, t_i]$.

Let $\delta\hat{\boldsymbol{e}}_{i\Delta T} \triangleq \hat{\boldsymbol{e}}_i - \boldsymbol{e}_i$ be the measured tracking error between the true system (4.2) against the nominal system (4.1) given the same initial condition, $\boldsymbol{e}_{i-1}$, and the same input over the interval, $\delta\hat{\boldsymbol{e}}_{i\Delta T}$ which can be decomposed as follows

$$\begin{aligned}\delta\hat{\boldsymbol{e}}_{i\Delta T} &= \int_{t_{i-1}}^{t_i} \left(\hat{f}(\hat{\boldsymbol{e}}(\tau)) - g_p(\boldsymbol{e}(\tau))\delta\boldsymbol{u}_{i-1}\right) d\tau + \underbrace{\delta\boldsymbol{e}_{(i-1)\Delta T}}_{=0} \\ &= \int_{t_{i-1}}^{t_i} g_p(\boldsymbol{e}(\tau))\boldsymbol{d}_{\boldsymbol{u}} d\tau + \int_{t_{i-1}}^{t_i} \boldsymbol{d}_{\boldsymbol{u}^\perp} d\tau,\end{aligned} \tag{4.16}$$

where $\boldsymbol{d}_{\boldsymbol{u}} \in \mathbb{R}^2$ represents the matched discrepancy similar to an input disturbance. The error $\boldsymbol{d}_{\boldsymbol{u}^\perp} = d^\perp \hat{\boldsymbol{n}}_{g_p}^\perp \in \mathbb{R}^3$ lies in the null space of g_p, which is spanned by the unit vector $\hat{\boldsymbol{n}}_{g_p}^\perp$. Parameter $d^\perp \in \mathbb{R}$ is the magnitude of the unmatched disturbance. By numerical differentiation with $\Delta T \longrightarrow 0$, the perturbed polar coordinate error dynamics in continuous time becomes:

$$\dot{\hat{\boldsymbol{e}}} = \dot{\boldsymbol{e}} + \delta\dot{\boldsymbol{e}} = g_p(\hat{\boldsymbol{e}})(\delta\boldsymbol{u} + \boldsymbol{d}_{\boldsymbol{u}}) + d_\perp \hat{\boldsymbol{n}}_{g_p}^\perp \tag{4.17}$$

with initial condition $\boldsymbol{e}(0) = \hat{\boldsymbol{e}}(0)$. For a physical robotic system, the expected dynamics discrepancy is bounded as stated in the following assumption.

Assumption 3. *The input (matched) disturbance $\boldsymbol{d}_{\boldsymbol{u}} : \hat{\mathcal{D}}^{\boldsymbol{e}} \longrightarrow \mathbb{R}^2$ and the unmatched drift (unmatched) $d_\perp : \hat{\mathcal{D}}^{\boldsymbol{e}} \longrightarrow \mathbb{R}$ are bounded.*

For safety, the system must be robust up to a user-defined risk tolerance against the worst-case discrepancies. Therefore, it is desirable to identify the matched and unmatched discrepancy bounds for both $||\boldsymbol{d}_{\boldsymbol{u}}||$ and $|d_{\perp}|$ using conformal prediction from the finite training data. Despite [108] addresses that conformal prediction can produce the desired minimum model discrepancy upper bounds (4.10), the assumption that the sampled nonconformity score satisfying (2.11) is not necessary valid given expected non-zero model residuals. Nevertheless, conformal prediction set coverage guarantees to hold regardless of the accuracy of the sampled nonconformity scores computed assuming zero model discrepancy, as they are designed for uncalibrated models [159].

Specifically, the desired probabilistic bounds on the matched and unmatched disturbances should satisfy the following:

$$Z_{\epsilon} \geq \min_{z\in\mathbb{R}}\{\mathbb{P}_{\hat{\Omega}}[|||\boldsymbol{d}_{\boldsymbol{u}}(E)|| \leq z] \geq 1-\epsilon\}, \tag{4.18}$$

$$Z_{\epsilon}^{\perp} \geq \min_{z\in\mathbb{R}}\{\mathbb{P}_{\hat{\Omega}}[|d_{\perp}(E)| \leq z] \geq 1-\epsilon\}, \tag{4.19}$$

where $\epsilon \in (0,1)$ is the user-defined risk for all random variables $E \in \hat{\Omega}$. Physically speaking, E denotes a random sample of polar tracking error over the set of all possible polar tracking errors represented as Ω. The following subsection details how to extract $||\boldsymbol{d}_{\boldsymbol{u}}||$ and $|d_{\perp}|$ from the training data and how to calculate Z_{ϵ} and $Z_{\epsilon}^{\perp}$ using conformal prediction.

Conformal-Driven Discrepancies

For each training state-input tuple, a tracking error $\delta\hat{\boldsymbol{e}}_{i\Delta T}$ is computed to obtain an estimate of the discrepancy bounds, $||\boldsymbol{d}_{\boldsymbol{u}}^{i}||$ and $||d_{\perp}^{i}||$. Specifically, the discrepancy bounds given samples i are the solution to the following equations in the given order:

$$||\boldsymbol{d}_{\boldsymbol{u}}^{i}|| = \max_{\boldsymbol{d}_{u}\in\mathbb{R}^2} \left\| |\delta\hat{\boldsymbol{e}}_{i\Delta T}| - \left| \int_{t_{i-1}}^{t_i} g_p(\boldsymbol{e}(\tau)) d\tau \boldsymbol{d}_{\boldsymbol{u}} \right| \right\|_2, \tag{4.20}$$

$$|d_{\perp}^{i}| = \left\| |\delta\hat{\boldsymbol{e}}_{i\Delta T}| - \left| \int_{t_{i-1}}^{t_i} g_p(\boldsymbol{e}(\tau)) d\tau \boldsymbol{d}_{\boldsymbol{u}}^{i} \right| \right\|_2. \tag{4.21}$$

Since input disturbance $\boldsymbol{d}_{\boldsymbol{u}}$ can be actively compensated by control input $\boldsymbol{u}$, the largest possible input discrepancy is prioritized and the remaining model mismatches will be allocated to the unmatched drift.

Assume that the training data is sufficiently rich. A L number of $||\boldsymbol{d}_{\boldsymbol{u}}^{i}||$ and $||\boldsymbol{d}_{\boldsymbol{u}^{\perp}}^{i}||$ is randomly sub-sampled without replacement where $0 \ll L < N_T$. Algorithm is developed which 2 uses training data set S_{N_T} to identify a $(1-\epsilon)$ upper bound $(Z_\epsilon, Z_\epsilon^\perp)$ defined in (4.18), (4.19) using the non-conformity scores $\{\|\boldsymbol{d}_{\boldsymbol{u}}^i\|\}_{i=0}^{L-1}$ and $\{\|\boldsymbol{d}_{\boldsymbol{u}^\perp}^i\|\}_{i=0}^{L-1}$.

Algorithm 2: Conformal-Driven Discrepancies

```
Data: Risk $\epsilon$, sub-sample $L$, training set $\{S_i\}_{i=1}^{N_T}$.
Result: $Z_\epsilon, Z_\epsilon^\perp$ defined by (4.18)-(4.19).
Compute index $q_\epsilon = \lceil (L+1)(1-\epsilon) \rceil$.
for $j$ from 1 to $L$ do
    Sample $\tilde{\boldsymbol{e}}$ without replacement and use (4.20) and (4.21) to compute $\|\boldsymbol{d}_{\boldsymbol{u}}^j\|$, $|d_\perp^j|$.
end
Add $\infty$ to the non-conformity scores $\{\|\boldsymbol{d}_{\boldsymbol{u}}^j\|\}_{j=0}^{L-1}$ and $\{\|\boldsymbol{d}_{\boldsymbol{u}^\perp}^j\|\}_{j=0}^{L-1}$.
Sort $\{\|\boldsymbol{d}_{\boldsymbol{u}}^j\|\}_{j=1}^{L+1}$ and $\{|d_\perp^j|\}_{j=1}^{L+1}$ in a non-decreasing orders.
$Z_\epsilon, Z_\epsilon^\perp$ are the $q_\epsilon^{th}$ smallest value in the sorted lists.
```

Implicitly, the choice of the non-conformity score assumes the discrepancy noise is uncolored and zero mean. Such an assumption always leads to more conservative estimates of the discrepancy upper bound for safety-critical applications. If the noise is known to be colored, a new nonconformity score that offsets the expected discrepancies, $\|\tilde{f}(X) - \mathbb{E}_\Omega[\tilde{f}(X)]\|_2$, could lead to less conservative identified upper bounds.

In summary, a distribution-free method is introduced to identify the probabilistic upper bounds for closed-loop tracking input discrepancies Z_ϵ and the unmatched drift $Z_\epsilon^\perp$. The following section augments the ancillary controller using the data-driven discrepancy bounds to identify the associated maximum trajectory tracking error.

4.3 Maximum Tracking Error Tubes

In addition to identified error dynamics (4.17), this section introduces the following perturbed variations of the nominal error dynamics (4.7) as intermediate cases to study to problem in the presence of both matched and unmatched disturbances,

$$\dot{\tilde{\boldsymbol{e}}} = g_p(\tilde{\boldsymbol{e}})(\delta\boldsymbol{u} + \boldsymbol{d}_{\boldsymbol{u}}), \tag{4.22}$$

$$\dot{\overline{\boldsymbol{e}}} = g_p(\overline{\boldsymbol{e}})\delta\boldsymbol{u} + d^\perp \hat{\boldsymbol{n}}_{g_p}^\perp, \tag{4.23}$$

where $\boldsymbol{e}(0) = \tilde{\boldsymbol{e}}(0) = \overline{\boldsymbol{e}}(0) = \hat{\boldsymbol{e}}(0)$. The perturbed dynamics (4.22) corresponds to the case where there are no unmatched discrepancies, *i.e.*, $d^\perp(t) = 0, \forall t$. Similarly, the dynamics of $\overline{\boldsymbol{e}}$ corresponds to the case where there is no matched uncertainty along the controllable subspace, *i.e.*, $\boldsymbol{d_u}(t) = 0, \forall t$. To study the effect of the discrepancies, the following lemmas are provided which leverage ideas from input-to-state stability.

Lemma 2. *Consider the perturbed error dynamics* (4.22)*. Assume a positive constant Z_ϵ exists and satisfies Eq.* (4.18) *for $t \in [0, \Delta T]$ as $\epsilon \to 0$. Define the set:*

$$\Omega_{ISS} = \left\{ \tilde{\boldsymbol{e}} \in \hat{\mathcal{D}}^{\tilde{\boldsymbol{e}}} \,\middle|\, V(\tilde{\boldsymbol{e}}) \le Z_\epsilon^2/4 \right\}, \tag{4.24}$$

where V is the positive definite function (4.8)*. Consider the augmented ancillary control law,*

$$\kappa_{ISS}(\tilde{\boldsymbol{e}}) = \kappa(\tilde{\boldsymbol{e}}) - \frac{1}{\lambda_1} g_p(\tilde{\boldsymbol{e}})^T \tilde{\boldsymbol{e}}. \tag{4.25}$$

For any $\tilde{\boldsymbol{e}}(0) \in \Omega_{ISS}$, if there exists a $\lambda_1 > 0$ such that applying the augmented control law (4.24) *yields the inequality $-\alpha_3(\alpha_2^{-1}(V(\tilde{\boldsymbol{e}}))) + \lambda_1 V(\tilde{\boldsymbol{e}}) \le -\tilde{\alpha}_3(V(\tilde{\boldsymbol{e}}))$, where $\tilde{\alpha}_3$ is a class $\mathcal{K}_\infty$ function, for all $\tilde{\boldsymbol{e}} \in \Omega_{ISS}$, then $\psi_{\tilde{\boldsymbol{e}}}(t, \kappa_{ISS}(\tilde{\boldsymbol{e}}), \tilde{\boldsymbol{e}}(0)) \in \Omega_{ISS}(t)$* [2] *for all $t \in (0, \Delta T)$.*

Proof. From the definition of the positive definite function V (4.8) and with the augmented controller κ_{ISS}, the following inequalities holds

$$\begin{aligned}
\dot{V} &\le -\alpha_3(\|\tilde{\boldsymbol{e}}\|) - \frac{1}{\lambda_1}\tilde{\boldsymbol{e}}^T g_p(\tilde{\boldsymbol{e}}) g_p(\tilde{\boldsymbol{e}})^T \tilde{\boldsymbol{e}} + \tilde{\boldsymbol{e}}^T g_p(\tilde{\boldsymbol{e}})\boldsymbol{d_u} \\
&\le -\alpha_3(\|\tilde{\boldsymbol{e}}\|) - \frac{1}{\lambda_1}\|g_p(\tilde{\boldsymbol{e}})^T \tilde{\boldsymbol{e}}\|_2^2 + |\tilde{\boldsymbol{e}}^T g_p(\tilde{\boldsymbol{e}})\boldsymbol{d_u}| \\
&\le -\alpha_3(\|\tilde{\boldsymbol{e}}\|) - \frac{1}{\lambda_1}\left\| g_p^T \tilde{\boldsymbol{e}} + \frac{\lambda_1}{2}\boldsymbol{d_u}\right\|_2^2 + \frac{\lambda_1 (Z_\epsilon)^2}{4} \\
&\le -\alpha_3(\|\tilde{\boldsymbol{e}}\|) + \frac{\lambda_1 (Z_\epsilon)^2}{4}.
\end{aligned}$$

Given $\lambda_1 > 0$ exists, the following inequalities hold

$$\begin{aligned}
\dot{V} &\le -\alpha_3(\|\tilde{\boldsymbol{e}}\|) + \lambda_1 V(\tilde{\boldsymbol{e}}) \\
&\le -\alpha_3(\alpha_2^{-1}(V(\tilde{\boldsymbol{e}})) + \lambda_1 V(\tilde{\boldsymbol{e}}) \le -\tilde{\alpha}_3(V(\tilde{\boldsymbol{e}}))
\end{aligned}$$

given perturbed dynamics (4.22). With $V(\boldsymbol{e}(0)) \le \frac{Z_\epsilon^2}{4}$, it can be shown that $V(\tilde{\boldsymbol{e}}(t)) \le \frac{Z_\epsilon^2}{4}, \forall t$, *i.e.*, $\psi_{\tilde{\boldsymbol{e}}}(t, \kappa_{ISS}(\tilde{\boldsymbol{e}}), \tilde{\boldsymbol{e}}(0)) \in \Omega_{ISS}$. □

[2] $\psi_{\tilde{\boldsymbol{e}}}(t, \kappa_{ISS}(\tilde{\boldsymbol{e}}), \tilde{\boldsymbol{e}}(0))$ is the solution (flow) of $\tilde{\boldsymbol{e}}$ governed by eqn. (4.22) under the augmented control law κ_{ISS} with the initial value $\psi(0) = \tilde{\boldsymbol{e}}(0)$.

It is important to recognize the set Ω_{ISS} is a Robust Control Invariant (RCI) set defined in section 2.4. Lemma 2 only considers the case where all disturbances lie in the controllable subspace. For the complementary case governed by (4.23), the Gronwall-Bellman inequality [160], is used to deduce the following result.

Lemma 3. *Under the nominal dynamical system* (4.7)*, the perturbed dynamical system* (4.23) *controlled by the nominal feedback control law* $\boldsymbol{u} = \kappa(\boldsymbol{e})$ *in Eqn.* (4.6) *has the closed-loop dynamics*

$$\begin{aligned}\dot{\boldsymbol{e}} &= g_p(\boldsymbol{e})\kappa(\boldsymbol{e}) = f_{cl}(\boldsymbol{e}),\\ \dot{\overline{\boldsymbol{e}}} &= g_p(\overline{\boldsymbol{e}})\kappa(\overline{\boldsymbol{e}}) + d^{\perp}\hat{\boldsymbol{n}}^{\perp}_{g_p} = f_{cl}(\overline{\boldsymbol{e}}) + d^{\perp}\hat{\boldsymbol{n}}^{\perp}_{g_p}.\end{aligned}$$

Let $Z^{\perp}_{\epsilon} \in \mathbb{R}_{>0}$ *satisfy* (4.19)*. Suppose there exists a Lipschitz constant* l_{cl} *with respect to the domain* $\hat{\mathfrak{D}}^{\boldsymbol{e}}$ *for function* f_{cl}*. Let* $\psi_{\boldsymbol{e}}(t, \kappa(\boldsymbol{e}), \boldsymbol{e}(0))$ *and* $\psi_{\overline{\boldsymbol{e}}}(t, \kappa(\overline{\boldsymbol{e}})), \overline{\boldsymbol{e}}(0))$ *be the flows of the systems* (4.7) *and* (4.23)*. The following inequality holds for all* $t \in [0, \Delta T]$*:*

$$\begin{aligned}&\|\psi_{\boldsymbol{e}}(t, \kappa(\boldsymbol{e}), \boldsymbol{e}(0)) - \psi_{\overline{\boldsymbol{e}}}(t, \kappa(\overline{\boldsymbol{e}})), \overline{\boldsymbol{e}}(0))\| \leq \\ &\qquad Z^{\perp}_{\epsilon} t e^{l_{cl}t} + \|\boldsymbol{e}(0)) - \overline{\boldsymbol{e}}(0))\| e^{l_{cl}t}. \qquad (4.26)\end{aligned}$$

Proof. By forward integration, the flows satisfy

$$\begin{aligned}&\|\psi_{\boldsymbol{e}}(t, \kappa(\boldsymbol{e}), \boldsymbol{e}(0)) - \psi_{\overline{\boldsymbol{e}}}(t, \kappa(\overline{\boldsymbol{e}})), \overline{\boldsymbol{e}}(0))\| \\ &\leq \|\boldsymbol{e}(0)) - \overline{\boldsymbol{e}}(0))\| + \int_0^t \left\| f_{cl}(\boldsymbol{e}) - f_{cl}(\overline{\boldsymbol{e}}) - d^{\perp}\hat{\boldsymbol{n}}^{\perp}_{g_p} \right\| d\tau \\ &\leq \|\boldsymbol{e}(0)) - \overline{\boldsymbol{e}}(0))\| + \int_0^t l_{cl} \|\boldsymbol{e}(\tau) - \overline{\boldsymbol{e}}(\tau)\| d\tau + Z^{\perp}_{\epsilon} t.\end{aligned}$$

The Gronwall-Bellman Inequality [160] yeilds (4.26). □

Intuitively speaking, the result of Lemma 6 states that if there are uncontrollable drift present, then the closed-loop system trajectory between the perturbed and nominal system grows exponentially in time. Building on Lemma 2 and 6, the stability and robustness in the presence of both input discrepancy and uncontrollable drift is studied next. As the main contribution, an augmented ancillary controller is derived which ensured closed-loop stability in the presence of the identified matched uncertainties and the associated maximum discrepancy tube for the full perturbed system (4.17).

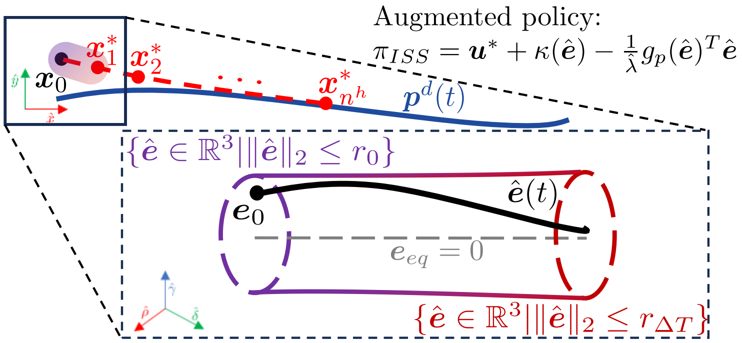
Augmented policy:
$\pi_{ISS} = \boldsymbol{u}^* + \kappa(\hat{\mathbf{e}}) - \frac{1}{\lambda} g_p(\hat{\mathbf{e}})^T \hat{\mathbf{e}}$
$\boldsymbol{x}_0$
$\boldsymbol{x}_1^*$
$\boldsymbol{x}_2^*$
$\boldsymbol{x}_{n^h}^*$
$\boldsymbol{p}^d(t)$
$\{\hat{\mathbf{e}} \in \mathbb{R}^3 \| \|\hat{\mathbf{e}}\|_2 \leq r_0\}$
$\boldsymbol{e}_0$
$\hat{\mathbf{e}}(t)$
$\boldsymbol{e}_{eq} = 0$
$\{\hat{\mathbf{e}} \in \mathbb{R}^3 \| \|\hat{\mathbf{e}}\|_2 \leq r_{\Delta T}\}$

Figure 4.4: Given reference path $\boldsymbol{p}^d(t)$ and optimal trajectory $\{\boldsymbol{x}_k^*, \boldsymbol{u}_k^*\}_{k=1}^{n_h}$, the maximum tracking error under augmented policy π_{ISS} is depicted. By converting the optimal state $\boldsymbol{x}_1^*$ and current state $\boldsymbol{x}_0$ into polar coordinate error $\boldsymbol{e}_0$, the tracking problem reduces to the stabilization of the error $\boldsymbol{e}_{eq} = 0$. Theorem 3 proves that if $\|\boldsymbol{e}_0\|_2 \leq r_0$, then the maximum tracking deviation during $t \in [0, \Delta T]$ is $\sup_{t\in[0,\Delta T]} \|\boldsymbol{e}(t)\|_2$ is upper bounded by $r_{\Delta T}$.

Theorem 3. *Suppose the system* (4.7) *is perturbed by input disturbance $\boldsymbol{d}_{\boldsymbol{u}}$ and uncontrolled drift $d_\perp$, as in* (4.17)*, where the disturbances $\boldsymbol{d}_{\boldsymbol{u}}$ and $d_\perp$ satisfy Assumption 3 given domain $\mathcal{D}^{\boldsymbol{e}}$. Given $\epsilon \to 0$, let Z_ϵ and $Z_\epsilon^\perp$ for $t \in [0, \Delta T]$ being the discrepancy upper bounds that also satisfy* (4.18) *and* (4.19)*, respectively.*

Let l_V be the Lipschitz constant for the function V in Eqn. (4.8) *on the bounded set $\hat{\mathcal{D}}^{\boldsymbol{e}}$. Let ΔT be small enough to satisfy*

$$Z_\epsilon^\perp \Delta T \exp(l_V \Delta T) \leq \alpha_1.$$

Consider a time varying radius $r(\tau)$ that satisfies

$$r(\tau) = r_\tau \triangleq \frac{(Z_\epsilon)^2}{4(\alpha_1 - Z_\epsilon^\perp \tau e^{l_V \tau})}. \tag{4.27}$$

Consider the augmented ancillary controller (4.25) *and error dynamics*

$$\dot{\boldsymbol{e}} \triangleq g_p(\boldsymbol{e})(\kappa_{ISS}(\boldsymbol{e}) + \boldsymbol{d}_{\boldsymbol{u}}),$$

$$\dot{\hat{\boldsymbol{e}}} \triangleq g_p(\hat{\boldsymbol{e}})(\kappa_{ISS}(\hat{\boldsymbol{e}}) + \boldsymbol{d}_{\boldsymbol{u}}) + d^\perp \hat{\boldsymbol{n}}_{g_p}^\perp,$$

where $\boldsymbol{d}_{\boldsymbol{u}}(0) = 0$, $d_\perp(0) = 0$ and $\|\boldsymbol{e}(0)\| \leq r(0)$. The existence of a $\hat{\lambda}_1 > 0$ such that $-\alpha_3(\alpha_2^{-1}(V(\hat{\boldsymbol{e}}))) + \hat{\lambda}_1 V(\hat{\boldsymbol{e}}) \leq -\hat{\alpha}_3(V(\hat{\boldsymbol{e}}))$ is true for all $t \in [0, \Delta T]$, where $\hat{\alpha}_3$ is a class $\mathcal{K}_\infty$ function, implies that the flow of the closed-loop [3] *tracking error $\hat{\boldsymbol{e}}(t)$ is bounded by $\|\hat{\boldsymbol{e}}(t)\| \leq r(t)$ over $t \in [0, \Delta T]$.*

[3] Referring to the error dynamics under augmented control law $\boldsymbol{u} = \kappa_{ISS}(\hat{e})$ (4.25) with $\hat{\lambda}_1$ replacing λ_1.

Proof. The continuous and differentiable function $\dot{V}$ is bounded over compact set $\hat{\mathcal{D}}^{\boldsymbol{e}}$ from Assumption 3. Therefore, V is Lipschitz continuous over domain $\hat{\mathcal{D}}^{\boldsymbol{e}}$. If there exist $\tilde{\lambda}_1 > 0$, then under κ_{ISS} yields

$$\dot{V}(\hat{e}) \leq -\alpha_3(\|\hat{\boldsymbol{e}}\|) + \frac{\hat{\lambda}_1 Z_\epsilon^2}{4} + |\hat{\boldsymbol{e}}^T \boldsymbol{d}_{\boldsymbol{u}}^\perp| \leq -\hat{\alpha}_3(V(\hat{\boldsymbol{e}})) + \|\hat{\boldsymbol{e}}\|_2 Z_\epsilon^\perp.$$

The first inequality follows the proof of Lemma 2 with the addition of $\hat{\boldsymbol{e}}^T \boldsymbol{d}_u^\perp \leq |\hat{\boldsymbol{e}}^T \boldsymbol{d}_u^\perp|$. The second inequality holds from a combination of the existence of $\hat{\lambda}_1$, the definition of $Z_\epsilon^\perp$ (4.19), and Holder's inequality.

From Lemma 6, the flow of the Lyapunov function V satisfies

$$\sup_{\tau\in[0,t]} |V(\boldsymbol{e}(\tau)) - V(\hat{\boldsymbol{e}}(\tau))| \leq \left(\sup_{\tau\in[0,t]} \|\hat{\boldsymbol{e}}\|_2 \, Z_\epsilon^\perp \tau \right) e^{l_V \tau},$$

where $|V(\boldsymbol{e}(0)) - V(\hat{\boldsymbol{e}}(0))| = 0$ because $\boldsymbol{e}(0) = \hat{\boldsymbol{e}}(0)$. By adding and subtracting zero and triangle inequalities, the inequalities hold

$$V(\hat{\boldsymbol{e}}(\tau)) \leq |V(\hat{\boldsymbol{e}}(\tau))| \leq |V(\boldsymbol{e}(\tau))| + |V(\hat{\boldsymbol{e}}(\tau)) - V(\boldsymbol{e}(\tau))|.$$

Recall the robust control invariance set (4.24). Let the initial error radius $r(0)$ take the explicit value

$$r(0) = \frac{(Z_\epsilon)^2}{4\alpha_1^{-1}}. \tag{4.28}$$

Further, since $\|\boldsymbol{e}(0)\| \leq r(0)$ and there exists a $\hat{\lambda}_1 > 0$, the unperturbed error dynamics satisfies $\|\boldsymbol{e}(t))\| \leq r(0)$ for all $t \in [0, \Delta T]$, by Lemma 5. The following inequalities can be derived by combining these inequalities:

$$\sup_{\tau\in[0,t]} V(\hat{\boldsymbol{e}}(\tau)) \leq \alpha_1 r_0 + \left(\sup_{t\in[0,\Delta T]} \|\hat{\boldsymbol{e}}(t)\| Z_\epsilon^\perp \Delta T \right) e^{l_V \Delta T} \tag{4.29}$$

because $V(\hat{\boldsymbol{e}}) \geq \alpha_1 \|\hat{\boldsymbol{e}}\|$. Substituting this inequality into equation (4.29), the following inequality holds

$$\sup_{\tau\in[0,t]} \|\hat{\boldsymbol{e}}(\tau)\| \leq r_0 + \sup_{\tau\in[0,t]} \|\hat{\boldsymbol{e}}(\tau)\| \frac{Z_\epsilon^\perp \tau e^{l_V \tau}}{\alpha_1}. \tag{4.30}$$

Lastly, combining equations (4.30) and (4.28) leads to the desired maximum tracking error bound (4.27). □

The increasing tube (radius-wise) defined by time-dependent radius $r(t)$ (4.27) in the interval $t \in [0, \Delta T]$ is the maximum tracking error tube, illustrated in Figure 4.4. Note, every planned desired trajectory $\boldsymbol{x}^*(t)$ resulting from desired inputs $\boldsymbol{u}^*(t)$ starting at state $\boldsymbol{x}_0$ is equivalent to stabilizing the tracking error from $\boldsymbol{e}_0$ to $[0, 0, 0]^T$. Leveraging the result of Theorem 3, the perturbed model tracking error $e(t)$ within each planning interval $t \in [0, \Delta T]$ will be inside a growing tube along the planned trajectory $\boldsymbol{x}^*(t)$ for $t \in [0, \Delta T]$ under the augmented controller κ_{ISS}, if the initial tracking error $\boldsymbol{e}_0$ is inside a ball of radius r_0 (4.28), as illustrated in Fig. 4.4.

Remark 2. *The condition* $\epsilon \longrightarrow 0$ *can lead to* $Z_\epsilon, Z_\epsilon^\perp \longrightarrow \infty$ *which is unpractical for robotic applications. Therefore for* $\epsilon \in (0, 1)$*, one can translate the result of Theorem 3 as the inequality* (4.27) *is true for* $(1 - \epsilon)$ *confidence level.*

4.4 Discrepancy-Aware Planning

Because differential-drive ground vehicles are underactuated, the unmatched uncertainties cannot be compensated in a controller synthesize problem. Nevertheless, the closed-loop trajectory drift (Lemma 6) can be effectively modulated through trajectory re-planning such as through heading adjustments, as demonstrated by the tube-MPC method [161].

To complete the framework, this section proposes a novel Discrepancy-Aware Planning algorithm that optimizes reference tracking while providing safety constraint satisfaction at a desired risk tolerance. There are two challenges studied: integrating maximum tracking error into the occupancy map to infer probabilistic safe traversal and finding optimal trajectories and inputs for the policy (2.43).

Probabilistic Safety-Critical Planning

Given a desired trajectory, the optimal tracking input can be found using a deterministic FTOCP with the following cost:

$$\mathcal{L}_{track} = \mathcal{L}_T(\delta \boldsymbol{p}_{t+n_h}) + \sum_{k=0}^{n^h-1} \left(\mathcal{L}_s(\delta \boldsymbol{p}_k) + \frac{1}{2} \boldsymbol{u}_k^T \boldsymbol{R} \boldsymbol{u}_k \right) \tag{4.31}$$

where $\delta \boldsymbol{p}_k = \boldsymbol{p}_k - \boldsymbol{p}_k^d$ is the output (position) tracking error in the inertial frame. Functions $\mathcal{L}_T \triangleq \delta \boldsymbol{p}_{t+n^h}^T Q_T \delta \boldsymbol{p}_{t+n^h}$ and $\mathcal{L}_s \triangleq \delta \boldsymbol{p}_k^T Q \delta \boldsymbol{p}_k$ for $k \in \{0, \ldots, n_h - 1\}$ are convex terminal and stage costs, respectively. Matrices Q_T, Q, $R \in \mathbb{R}^{2 \times 2}$ are positive definite. Note, maximum tracking cost threshold, $\overline{\mathcal{L}}_{track} \in \mathbb{R}_{>0}$ is applied. The parameter $\overline{\mathcal{L}}_{track}$ stops the cost-to-go from becoming unbounded as the output tracking error increases as $\mathcal{L}_{track} = \min(\mathcal{L}_{track}, \overline{\mathcal{L}}_{track})$. The authors infer $\overline{\mathcal{L}}_{track}$

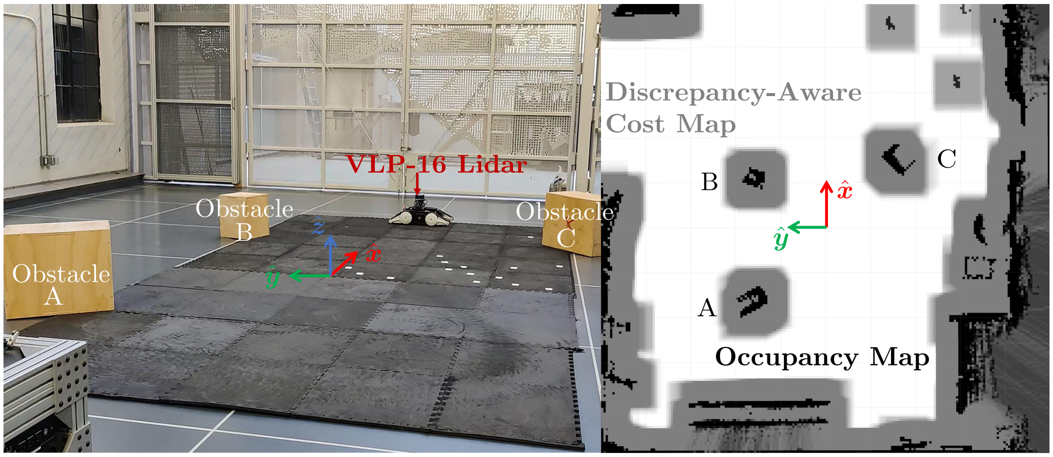

Figure 4.5: Experimental Set up Schematics. (Left) Configuration B test set up capture where three stationary obstacles are placed such that the vehicle needs to deviate from its reference trajectory to avoid. (Right) An overlay of the raw occupancy map and the discrepancy-aware cost map C_ϵ with $\epsilon \in (0.0013, 0.0016)$ resulting a $N_\epsilon = 16$ composed by $r_{ego} = 0.40$ m and $r_{buff} = 0.15$ m with $r_{map} = 0.05$ m.

from the system's input limits and the planning horizon such that the maximum stage cost corresponds to the physical limits imposed by the robotic hardware.

Given a vehicle operating state X_t and its projected nominal state $\boldsymbol{x}_t$, the solution to the following discrete-time optimization problem yields the desired probabilistically safe input sequence

$$\{\boldsymbol{u}^*_{t+k}\}_{k=0}^{n_h-1} = \min_{\{\boldsymbol{u}_{t+k}\}_{k=0}^{n_h-1}} \sum_{k=0}^{n_h-1} \mathcal{L}_{track} \tag{4.32a}$$

$$\text{s.t.} \quad X_{t+k+1} = \hat{f}^d_{t+k}(X_{t+k}) \tag{4.32b}$$

$$\boldsymbol{x}_{t+k+1} = P_{X\to\boldsymbol{x}}(X_{t+k+1}) \in \mathcal{D}^{\boldsymbol{u}} \tag{4.32c}$$

$$\boldsymbol{u}_{t+k} = P_{X\to\boldsymbol{u}}(X_{t+k}) \in \mathcal{D}^{\boldsymbol{u}}, \quad \forall k \in \mathbb{Z}_0^{n_h-1} \tag{4.32d}$$

$$\boldsymbol{x}_t = P_{X\to\boldsymbol{x}}(\boldsymbol{x}_t) \tag{4.32e}$$

$$\mathbb{P}_\Omega[M\boldsymbol{x}_{t+1} \in \overline{O}] \leq \epsilon, \tag{4.32f}$$

that avoids all occupied grids in the map O at a minimum probability $(1-\epsilon)$ percent. Equation (4.32b) represents a discrete-time update of the true system operating state X with time discretization step size ΔT. The matrix $M \in \mathbb{R}^{2\times 3}$ maps from nominal states $\boldsymbol{x}$ to vehicle positions $\boldsymbol{p}$, and constraint (4.32e) means the initial condition is set by the current state. Inequality (5.3d) constrains the probability of the planned vehicle's position, at time $t+1$ given the current nominal state $\boldsymbol{x}_t$, to be inside an augmented occupied set, denoted as $\overline{O} \subseteq \mathbb{R}^2$, is less than ϵ. The set $\overline{O}$ is constructed from O with the addition of the vehicle geometry such that if the point mass at position $\boldsymbol{p} \in \mathbb{R}^2$ satisfied $\boldsymbol{p} \notin \overline{O}$, then the vehicle occupied points given center position $\boldsymbol{p}$ denoted as the set $\mathcal{S}_{\boldsymbol{p}} \subset \mathbb{R}^2$ satisfied $\mathcal{S}_{\boldsymbol{p}} \cap O = \emptyset$. Physically speaking, the vehicle should not enter the obstacle-occupied grids given by the map O more frequently than the specified risk tolerance.

The optimization problem (4.32) is difficult to solve because the true system dynamics (4.32b) is unknown and unobservable, constraint (5.3d) is nondeterministic, and the map O outputs discontinuous values given continuous positions. As a solution, (4.32) is reformulated into a deterministic optimization program while providing the same or lower obstacle collision tolerance ϵ.

Recall that polar coordinates state ρ represents the displacement between the vehicle's actual position and the planned one. The result of Theorem 3 yields

$$\sup_{t\in[0,\Delta]} |\hat{\rho}(t)| = \sup_{t\in[0,\Delta T]} |\boldsymbol{i}^T \hat{\boldsymbol{e}}(t)| \leq \|\boldsymbol{i}\| \sup_{t\in[0,\Delta T]} \|\hat{\boldsymbol{e}}(t)\| \leq r_{\Delta T},$$

where $\boldsymbol{i} = [1, 0, 0]^T$. Given the current nominal state $\boldsymbol{x}_t$, if the initial tracking error arising from the planned state $\boldsymbol{x}^*_{t+1}$ satisfies $\|\boldsymbol{e}_0\| \leq r_0$, then this constraint guarantees up to $1 - \epsilon$ confidence that vehicle position $\boldsymbol{p}_{t+1}$, at time $t + \Delta t$, is an element of the following set

$$\mathcal{S}_{ISS} = \{\boldsymbol{p} \in \mathbb{R}^2 \mid \|\boldsymbol{p}_{t+1}(\boldsymbol{x}_t, \kappa(\boldsymbol{e}_0)) - \boldsymbol{p}^*_{t+1}\| \leq r_{\Delta T}\}. \tag{4.33}$$

That is, if the constraint $\|\boldsymbol{e}_0(\boldsymbol{x}_t, \boldsymbol{x}^*_{t+1})\| \leq r_0$ is satisfied, then enforcing an added distance gap of $r_{\Delta t}$ between the ego vehicle and the obstacle is sufficient to guarantee obstacle avoidance with a $(1 - \epsilon)$ confidence.

Theorem 4. *Consider a differential-driven UGV with nominal dynamics* (4.1)*, true dynamics* (4.2)*, and reference trajectory* $\boldsymbol{p}^d(t)$*. Denote the current vehicle operating state as* X_t *and its projected nominal state as* $\boldsymbol{x}_t$*. The polar tracking error satisfies nominal dynamics* (4.7)*. Consider the tracking control policy* (2.43) *where the nominal ancillary controller* κ (4.6) *is used for stabilizing the tracking error* $\boldsymbol{e}$*.*

Suppose (4.7) *is perturbed by an additive matched disturbance,* $g_p(\boldsymbol{e})\boldsymbol{d_u}$*, and an additive unmatched model disturbance* $d_\perp \hat{\boldsymbol{n}}^\perp_{g_p}$ *to* (4.17)*, and there exist* $Z_\epsilon \in \mathbb{R}_{>0}$ *satisfying* (4.18) *and* $Z^\perp_\epsilon \in \mathbb{R}_{>0}$ *satisfying* (4.19) *given a risk tolerance* $\epsilon \in [0, 1]$*. Suppose there exists* $\tilde{\lambda}_1 \in \mathbb{R}_{>0}$ *in Theorem 3. Let* $\{\mathcal{FS}^*_{t+k}\}_{k=0}^{n_h-1}$ *be the set of* *admissible inputs* [4] *for the following optimization problem.*

$$\{\boldsymbol{u}^*_{t+k}\}_{k=0}^{n_h-1} = \min_{\{\boldsymbol{u}_{t+k}\}_{k=0}^{n_h-1}} \sum_{k=0}^{n_h-1} \mathcal{L}_{track} \tag{4.34a}$$

$$s.t. \quad \boldsymbol{x}_{t+k+1} = f^d_{t+k}(\boldsymbol{x}_{t+k}, \boldsymbol{u}_{t+k}, \Delta t) \tag{4.34b}$$

$$\boldsymbol{e}_k = f_{\boldsymbol{e}}(\boldsymbol{x}_{t+k}, \boldsymbol{x}_{t+k+1}) \tag{4.34c}$$

$$\|\boldsymbol{e}_k\| \leq r_0 \tag{4.34d}$$

$$C(\boldsymbol{p}_{t+k+1}, r_{(k+1)\Delta T}) \cap \overline{O} = \emptyset \tag{4.34e}$$

$$\boldsymbol{u}_{t+k} + \kappa_{ISS}(\boldsymbol{e}_k) \in \mathcal{D}^{\boldsymbol{u}}, \quad \forall k \in \mathbb{Z}_0^{n_h-1} \tag{4.34f}$$

$$\boldsymbol{x}_t = \boldsymbol{x}_t, \tag{4.34g}$$

where equality constraint (4.34b) *is the discrete-time version of nominal model* (4.1). *The set $C(\boldsymbol{p}_{k+1}, r_{\Delta T})$ in* (4.34e) *denotes the closed disc with center $\boldsymbol{p}_{k+1} \in \mathbb{R}^2$ and radius $r_{\Delta T(k+1)} \in \mathbb{R}_{>0}$. Let $\{\mathcal{FS}_k\}_{k=0}^{n_h-1}$ be the sequence of feasible input sets to the optimization problem* (4.32)*. If r_t is computed using* (4.27)*, then $\mathcal{FS}^*_{t+k} \subseteq \mathcal{FS}_{t+k}$, for all $k \in \mathbb{Z}_0^{n_h-1}$.*

Proof. As an overview, set inclusion is shown using a standard approach, showing every element of the set $\mathcal{FS}^*_k$ also belongs to the set $\mathcal{FS}_k$ for all $k \in \mathbb{Z}_0^{n_h-1}$. Consider $k = 0$. From Theorem 3, under the augmented ancillary controller κ_{ISS}, the position flow of the perturbed system $\boldsymbol{\psi}_{\boldsymbol{p}}(\tau)$ for $\tau \in [t, t+\Delta T]$ under any input $\boldsymbol{u}_t \in \mathcal{FS}^*_t + k$ with initial position $\boldsymbol{p}_t$ and $(1-\epsilon)$ confidence, satisfies the following,

$$\boldsymbol{\psi}_{\boldsymbol{p}}(\tau) - \boldsymbol{p}^*(\tau) \leq \sup_{\tau \in [t, t+\Delta T]} \|\delta \boldsymbol{p}(\tau)\| \leq \sup_{\tau \in [t, t+\Delta T]} |\rho(\tau)| \leq r_{\Delta T}.$$

The above inequality implies, for all $\boldsymbol{u}_t \in \mathcal{FS}^*_0$, satisfaction of the inequality constraints (4.34d) and (5.3c) is equivalent to the statement that the perturbed system's position flow $\boldsymbol{\psi}_{\boldsymbol{p}}(\tau)$ for $\tau \in [t, t+\Delta T]$ does not intersect an occupied grid with $(1-\epsilon)$ confidence. Therefore, control inputs $\boldsymbol{u}_t \in \mathcal{FS}^*_t$ are also feasible solutions to optimization problem (4.32) yielding $\boldsymbol{u}_t \in \mathcal{FS}_t$, meaning $\mathcal{FS}^*_0 \subseteq \mathcal{FS}_0$. For $k = 1$, the flow generated by the composite following input can be similarly inspected

$$\boldsymbol{u}(\tau) = \begin{cases} \boldsymbol{u}_t \in \mathcal{FS}^*_t, & \tau \in [t, t+\Delta T] \\ \boldsymbol{u}_{t+1} \in \mathcal{FS}^*_{t+1}, & \tau \in [t+\Delta T, t+2\Delta T]. \end{cases}$$

[4] If a input sequences $\{\boldsymbol{u}_{t+k}\}_{k=0}^{n_h-1}$ satisfies the constraints (4.34b) - (4.34f) given current state $\boldsymbol{x}_t$ then such input sequence is an element of the set $\{\mathcal{FS}^*_{t+k}\}_{k=0}^{n_h-1}$, $\boldsymbol{u}_{t+k} \in \mathcal{FS}^*_{t+k}$ for all $k \in \mathbb{Z}_{k=0}^{n_h-1}$.

Building on the result for $k = 0$, the position flow of the system satisfies $\boldsymbol{\psi}_{\boldsymbol{p}}(\tau) \leq \boldsymbol{p}^*(\tau)+r_{2\Delta T}$ for $\tau \in [t, t+2\Delta T]$ with confidence $1-\epsilon$. Therefore, $\forall \boldsymbol{u}_{t+1} \in \mathcal{FS}^*_{t+1}$ are also member of set $\mathcal{FS}_{t+1}$, yielding $\mathcal{FS}^*_{t+1} \subseteq \mathcal{FS}_{t+1}$. The proof for the remaining $k \in \mathbb{Z}^{n_h-1}_{k=2}$ follows by induction. □

Based on Theorem 4, solving the deterministic optimization problem (4.34) with an inaccurate but known nominal model guarantees the desired obstacle avoidance behavior from solving the original optimization problem (4.32). Lastly, the optimization-based planner is assumed to update and recompute at every ΔT seconds with latest state update such that constraints (4.34d) and (4.34d) are only applied at $k = 0$ [5].

Remark 3. *In Theorem 4, a circular robot shape* (4.34e) *can be replaced by tighter polytopic boundary. In this case, the cost map introduced in the following section may become orientation-dependent.*

Discrepancy-Aware MPPI

The nonlinear program (4.34) can still be numerically challenging to solve, especially with the set O represented as a grid map. Therefore, a discrepancy-aware cost map is proposed to encode the obstacle avoidance constraint (4.34e). This map facilitates cost minimization, ensuring robust obstacle avoidance amid model inaccuracy and vehicle geometry. Specifically, consider the following augmented cost function which combines the reference tracking cost (4.31) with a collision penalty,

$$\overline{\mathcal{L}} = \mathcal{L}_{track} + \sum_{k=1}^{n_h} \mathcal{L}_C(\boldsymbol{p}_k, C_\epsilon) + \alpha_{ISS}\mathbf{1}_{\|\boldsymbol{p}_{t+1}-\boldsymbol{p}_t\| \geq r_0}, \tag{4.35}$$

where $\mathcal{L}_C : \mathbb{R}^2 \times \mathbb{R}^{l_{map} \times w_{map}} \longrightarrow \mathbb{R}$ is a map induced cost function which takes in a position $\boldsymbol{p} \in \mathbb{R}^2$ and a $l_{map} \times w_{map}$ sized discrepancy aware cost map to produce a collision cost. The parameter $\alpha_{ISS} \in \mathbb{R}_{>0}$ is chosen to enforce the algorithm to plan a trajectory satisfying the requirement $\|\boldsymbol{e}_0\| \leq r_0$ [6].

Using the discrete-time nominal model (4.34b), MPPI is applied to derive sub-optimal trajectory and input sequences $(\{\boldsymbol{x}^*_{t+k}, \boldsymbol{u}^*_{t+k}\}_{k=1}^{n_h})$ using the importance sam-

[5] Enforcing trajectory level safety, by adding safety distance of $r_{(k+1)\Delta T}$ at future horizon steps, $k \in \mathbb{Z}_1^{n_h-1}$, can lead to overly conservative behavior.

[6] The term $\alpha_{ISS}\mathbf{1}_{\|\boldsymbol{p}_{t+1}-\boldsymbol{p}_t\| \geq r_0}$ can be dropped if sample trimming is incorporated where a MPPI sampled trajectory is omitted when the inequality $\|\boldsymbol{p}_{t+1} - \boldsymbol{p}_t\| \geq r_0$ holds. A replanning scheme instead of sample trimming is being implemented in the experimental validation section where if the inequality $\|\boldsymbol{p}_{t+1} - \boldsymbol{p}_t\| \geq r_0$ holds, a new set of input discrepancies are sampled until the converse is true.

pling aggregating law (2.35)-(2.36) and the proposed discrepancy-aware cost function (4.35). To manage input constraints, an element-wise clamping function $\boldsymbol{u}_{clamp} = \max(\min(\boldsymbol{u}_{max}, \boldsymbol{u}), \boldsymbol{u}_{min})$ is recommended, which does not affect the MPPI algorithm's convergence [31]. The initial input sequence (warm start) is obtained using the nominal dynamics and flatness properties as (4.5). Algorithm 3 details this process, including the construction of the cost map C_ϵ.

Discrepancy-Aware Cost Map

To account for the vehicle footprint in obstacle avoidance, given the nominal model (4.34b) focuses on the UGVs' center of mass, the smallest circumscribing circle's radius, r_{ego}, is computed see Fig. 4.6 which is used to enlarge each occupied grid by r_{ego} for center-focused obstacle avoidance. Additionally, to accommodate model mismatches, the occupied grids are also enlarged by a radius $r_{\Delta T}$, creating a "collision buffer" around each occupied grid. To encode this collision buffer on a nominal grid map, the positive integer $N_\epsilon \triangleq \lceil (r_{\Delta T} + r_{ego})/r_{map} \rceil$ denotes the buffer's grid size. The discrepancy-aware cost map can be interpreted as a N_ϵ grid "inflation" on top of the occupancy map as an additive safety buffer zone between the ego vehicle and surrounding obstacles, see Fig 4.6 as an example of such inflation. The construction of the cost map is detailed in Algorithm 3 where a description of the process is provided.

Discrepancy-Aware Cost Map Construction Details

For every occupied grid, its neighboring grids within a distance of $(r_{\Delta T} + r_{ego})$, referred to as the safety buffer, will also be considered as occupied grids to ensure robust obstacle avoidance given model inaccuracy and vehicle geometry.

An enlarged grid map $\hat{O} \in \mathbb{R}^{l_{map}+2N_\epsilon \times w_{map}+2N_\epsilon}$ is initialized, and the nominal map is shifted and aggregated to form the collision buffer grid map (4.36) and visualized in Figure 4.1. Specifically, the nominal occupancy map is shifted by N_ϵ grids in both column-wise and row-wise positive and negative directions, as described in Algorithm 3. In total, there are $4N_\epsilon + 2$ shifted grid maps which are grid-wise aggregated to obtain the grid map with collision buffer,

$$\hat{O}_{buffer} \triangleq \frac{1}{4N_\epsilon + 2} \sum_{i=-N_\epsilon}^{N_\epsilon} \sum_{j=-N_\epsilon}^{N_\epsilon} \frac{\hat{O}^s_{i,j}}{100}, \tag{4.36}$$

where $\hat{O}^s_{i,j} \triangleq \hat{O}(N_\epsilon + i : N_\epsilon + l_{map} + i - 1, N_\epsilon + j : N_\epsilon + w_{map} + j - 1)$ is an example grid shifting operation. The normalization factor 100 is applied on the enlarged occupancy maps to obtain a value within $[0, 1]$.

The discrepancy-aware cost map C_ϵ is constructed by applying an adjustable cost multiplier $\alpha_{shift}/N_{shift,\epsilon}$ to this enlarged occupancy map $\hat{O}_{buffer}$ The multiplier $\alpha_{shift}/N_{shift,\epsilon}$ to the N_ϵ-shifted occupancy maps is a design parameter where $\alpha_{shift} \in \mathbb{R}_{>0}$ is a cost multiplier similar to α_{ISS} in (4.35) and the positive integer $N_{shift} \in \{1, \ldots, N_\epsilon\}$ serves as a cost decay. This cost adjustment accounts for both conservative obstacle avoidance and model mismatches, ensuring that the true and uncertain system flow remains collision-free given the risk tolerance. The final cost map balances obstacle collision penalties and tracking costs (Algorithm 3). The design parameter $\alpha_{shift}/N_\epsilon$ needs to be lower bounded by the maximum tracking cost $\overline{L}_{track}$ to penalize unsafe trajectories. A larger α_{shift} leads to more conservative obstacle avoidance behavior, and a smaller α_{shift} reduces the cost associated with the avoidance of the safe buffer due to model mismatches. The discrepancy-aware cost map, denoted as C_ϵ (See Algorithm 3), is the centering l_{map} and w_{map} grids of the following enlarged cost map:

$$C_{buffer} \triangleq \sum_{i=-N_\epsilon}^{N_\epsilon} \sum_{j=-N_\epsilon}^{N_\epsilon} \alpha_{shift}/\underbrace{\sqrt{i^2+j^2}}_{\triangleq N_{shift,\epsilon}} \hat{O}^s_{i,j}. \tag{4.37}$$

Lastly, a maximum cost map threshold $C_\epsilon = \min(C_\epsilon, \overline{L}_{track})$ is enforced with the maximum tracking cost grid-wise to even the cost penalty for obstacle collision.

As a result, a discrepancy-aware cost map, denoted as C_ϵ, is constructed that satisfies the following property:

$$C\epsilon(\boldsymbol{p}) \geq \overline{\mathcal{L}}_{track} \quad \text{iff} \quad C(\boldsymbol{p}, r_{\Delta T} + r_{ego}) \cap \overline{O} \neq \emptyset \tag{4.38}$$

for all positions $\boldsymbol{p} \in \mathbb{R}^2$ that is within the range of the occupancy map.

For MPPI cost evaluation, the position $\boldsymbol{p} = [x, y]^T$ is mapped to the corresponding grid indices on $C\epsilon$, calculating $\mathcal{L}_C(\boldsymbol{p}, C\epsilon)$ as the cost map entry at those indices [7].

As an extension to the results in Theorem 3 and 4, one can make the following guarantees of the proposed data-driven planning and control framework.

Corollary 1. *Suppose sufficient training samples S, given in the form* (4.14)*, are available to obtain $Z_\epsilon, Z_\epsilon^\perp$ offline using Algorithm 2 given a user-defined risk-level*

[7] Explicitly, the index $X_C = \lfloor \frac{x}{r_{map}} + \frac{l_{map}}{2} \rfloor$ and $Y_C = \lfloor \frac{y}{r_{map}} + \frac{w_{map}}{2} \rfloor$ can be computed given $\boldsymbol{p}_k$. Then, $\mathcal{L}_C(\boldsymbol{p}_k, C_\epsilon)$ is the $(X_C, Y_C)^{th}$ entries of C_ϵ.

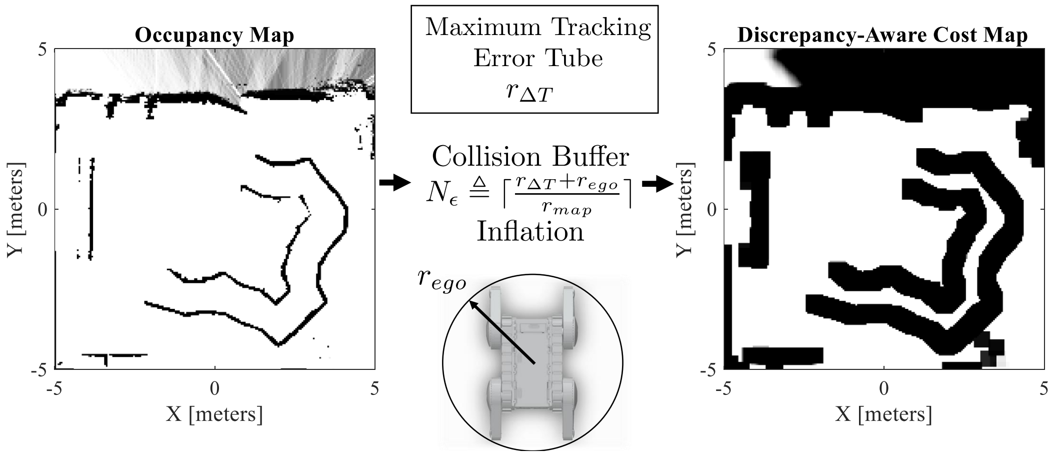
Occupancy Map
Y [meters]
X [meters]
5
0
-5
-5
0
5
Maximum Tracking Error Tube $r_{\Delta T}$
Collision Buffer $N_\epsilon \triangleq \lceil \frac{r_{\Delta T}+r_{ego}}{r_{map}} \rceil$ Inflation
r_{ego}
Discrepancy-Aware Cost Map
Y [meters]
X [meters]
5
0
-5
-5
0
5

Figure 4.6: A pictorial overview of the construction of the Discrepancy-Aware Cost Map.

$\epsilon \in (0, 1)$. *Suppose there exist* $\tilde{\lambda}_1 > 0$ *such that Theorem 3 holds. Under the same conditions and assumptions as Theorem 4, a local minimum trajectory* $\{\boldsymbol{x}^*_{t+k}\}_{k=1}^{n_h}$ *given by input sequence* $\{\boldsymbol{u}^*_{t+k}\}_{k=0}^{n_h-1}$ *is obtained using the discrepancy-aware MPPI algorithm, given by Algorithm 3. Applying control input*

$$\boldsymbol{u}_t = \boldsymbol{u}^*_t + \kappa_{ISS}(\boldsymbol{e}_0) \tag{4.39}$$

is sufficient to avoid the occupied grids in map O *with a minimum* $(1-\epsilon)$ *confidence, if the following conditions are satisfied:*

- *The control inputs* $\boldsymbol{u}_t$ *from Eqn.* (4.39) *satisfy input constraint, i.e.,* $\boldsymbol{u}_t \in \mathcal{D}_{\boldsymbol{u}}$.
- *The trajectory cost* $\overline{\mathcal{L}}(\{\boldsymbol{x}^*_k, \boldsymbol{u}^*_k\}_{k=0}^{n_h-1}) < \overline{\mathcal{L}}_{track}$.

Proof. Based on the cost assignment of C_ϵ, if the total cost of the planned trajectory $\overline{\mathcal{L}}(\{\boldsymbol{x}^*_k, \boldsymbol{u}^*_k\}_{k=0}^{n_h-1})$ is less than $\overline{\mathcal{L}}_{track}$, the inequality $\mathcal{L}_C(\boldsymbol{p}^*_{t+1}, C_\epsilon) \leq \overline{\mathcal{L}}_{track}$ is automatically guaranteed, *i.e.*, collision safety with confidence $1 - \epsilon$. Given the control policy (4.39) also satisfies the control input limit, the flow of the perturbed system (4.17) will be within an expanding maximum track error tube of radius r_t, calculated using (4.27), along the planned position trajectory $\{\boldsymbol{p}^*_{t+k}\}_{k=0}^{1}$ with $(1-\epsilon)$ confidence. □

The theoretical guarantees from Corollary 1 can serve as planning safety validation and verification in addition to synthesizing a safe controller as shown in the driver-assist application in Section 4.4. Most importantly, the MPPI planned trajectories always perform cost minimizing which returns to the collision-free regions even if there are no feasible safe input sequences, more convenient relative to optimization-based MPC algorithms which require back-up strategies when the optimization problem becomes infeasible.

Algorithm 3: Discrepancy-Aware MPPI

```
Data: Map parameters $N_\epsilon$, $r_{map}$, $x_{O,c}$, $y_{O,c}$, occupancy map $O$, current
      nominal state $\boldsymbol{x}_t$, goal position $\boldsymbol{p}_t^d$, cost function parameters: $Q$, $Q_T$, $R$,
      $\alpha_{cost}$, $\alpha_{shift}$, MPPI parameters $\Sigma_{\boldsymbol{u}}$, $N_{sample}$,$\lambda$,$\alpha_{ISS}$, initial control
      sequence $\{\overline{\boldsymbol{u}}\}_0^{n_h-1}$, and horizon $n_h$.
Result: $\{\boldsymbol{x}_i^*\}_{i=1}^{n_h}$, $\{\boldsymbol{u}_i^*\}_{i=0}^{n_h-1}$, $\boldsymbol{u}_{send}$
Create $l_{map} + 2N_\epsilon$ by $w_{map} + 2N_\epsilon$ grid map $\hat{C}_{buffer}$.
Create $l_{map} + 2N_\epsilon$ by $w_{map} + 2N_\epsilon$ enlarged occupancy map $\hat{O}$ based on $O$.
while task not completed do
    for $i = -N_\epsilon, i < N_\epsilon, i++$ do
        for $j = -N_\epsilon, j < N_\epsilon, j++$ do
            $C_{buffer}(N_\epsilon + i : N_\epsilon + l_{map} + i - 1, N_\epsilon + j : N_\epsilon + w_{map} + j - 1) += \frac{\alpha_{shift}\hat{O}^s_{i,j}}{100(\sqrt{i^2+j^2+1})}$.
        end
    end
    $C_\epsilon = \lceil C_{buffer}(N_\epsilon : N_\epsilon + l_{map} - 1, N_\epsilon : N_\epsilon + w_{map} - 1), \overline{\mathcal{L}}_{track} \rceil$.
    for $k = 0, k < N_{sample} - 1, k++$ do
        Draw $\delta\boldsymbol{u}$ from $\mathcal{N}(\boldsymbol{0}, \Sigma_{\boldsymbol{u}})$ $n_h$ times. for $i = 0, i < n_h - 1, i++$ do
            $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + g(\boldsymbol{x}_i)(\min(\max(\overline{\boldsymbol{u}}_i + \delta\boldsymbol{u}_i, \boldsymbol{u}_{min}), \boldsymbol{u}_{max})\Delta t$
        end
        Evaluate the $k^{th}$ MPPI trajectory cost $\overline{\mathcal{L}}_k$ using (4.35).
    end
    Compute the optimal MPPI input sequence $\{\boldsymbol{u}_i^*\}_{i=0}^{n_h-1}$ using equations
      (2.35)-(2.36) with $\overline{\mathcal{L}}_k$ as the flow costs.
    Compute best sampled state trajectory $\{\boldsymbol{x}_i^*\}_{i=1}^{n_h}$ and use $\boldsymbol{x}_1^*$ and $\boldsymbol{x}_0$ to
      compute polar tracking error $\hat{\boldsymbol{e}}$.
    Use (4.41) to compute the input that will be sent to actuators.
    For $i$=0 : $n_h - 2$: $\overline{\boldsymbol{u}}_i = \boldsymbol{u}_{i+1}^*$. Fill $\overline{\boldsymbol{u}}_{n_h-1}^*$ using flatness relationship (4.5).
end
```

4.5 Hardware Validation: Autonomy in Clustered Environment

The proposed framework is first validated through experiments focusing on high-speed trajectory tracking given stationary obstacles. These experiments are conducted using four ground and vehicle configurations, as depicted in Fig. 4.2.

Specifically, the Flipper Pro by Rover Robotics is chosen as the ego UGV which has

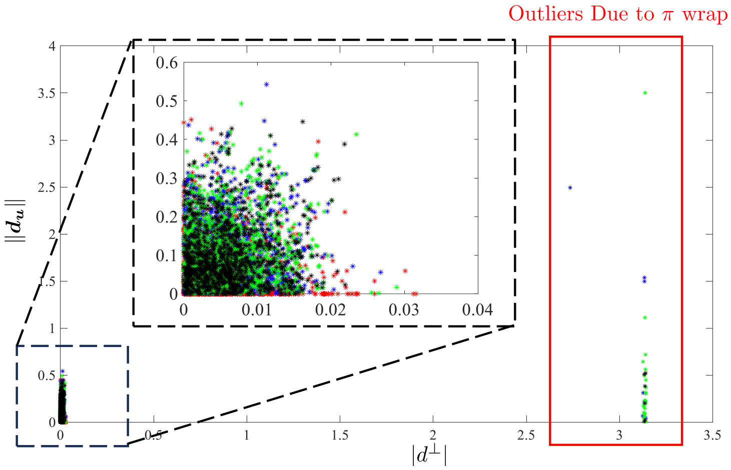

Figure 4.7: Data-driven discrepancy identification results from configuration B with $T_{lap} = \{20, 30, 40, 50\}$ seconds. From the training set, the absolute uncontrollable drift is plotted against the input discrepancy norm, observing non-negligible model discrepancy upper bounds. Outliers are observed due to incorrect π wrapping which can skew the empirical distribution formed by the nonconformity scores as well as the conformal-driven discrepancies.

$r_{ego} = 0.39$ m in flipper configuration and $r_{ego} = 0.3$ m in no flipper configuration. There are three stationary crates in the test field acting as obstacles, strategically placed to test the robot's obstacle avoidance capabilities (refer to Fig. 4.5 for layout details).

A 2D occupancy map is contracted which centers at $(0, 0)$ with a 0.05 m grid size, updated at 2 Hz, using a LiDAR sensor (VLP-16 by Velodyne) mounted on the robot. The standard occupancy grid mapping algorithm [162] is employed to construct the occupancy map with robot poses provided by an OptiTrack motion capture system. An input limit of $v_{max} = [-2, 2]$ m/s and $\omega_{max} = [-2, 2]$ rad/s is enforced. The dead zone is selected to be $\rho_{min} = 0.05$ m, and the maximum position error is set to $\rho_{max} = 0.5$ m, considering the input limits. All processes are executed by an onboard companion computer, powered by an AMD Ryzen^{TM} 9 6900HX CPU, within a ROS 1 environment.

Training Details

Given that both terrains exhibit uniform characteristics, model discrepancies are primarily coupled with linear and angular velocities and input time delays. To obtain training data that effectively captures these discrepancies, a "Figure 8" trajectory is tracked at four different desired lap timings without obstacles, denoted as T_{lap}, for all four configurations. Specifically, the trajectory as: $x^d(t) = 2.5\cos(2\pi t/T_{lap})$ and

$y^d(t) = 1.25\sin(4\pi t/T_{lap})$ is prescribed, where T_{lap} takes values of 20, 30, 40, and 50 seconds. The data collection process involves gathering approximately 5 minutes of data at each configuration, with a sampling rate of 20 Hz. A scatter plot of the identified unmatched and matched disturbances from the training data is displayed in Fig. 4.7 which showed non-negligible model discrepancies, mainly arising from input delays and track slipping. Subsequently, the conformal-driven upper bounds are computed to establish the maximum tracking deviation radii, presented in Table 5.1 using a sub-sample count of $L = 3000$.

Choice of Parameters

For the control algorithm, a frequency of 20 Hz was selected. the MPPI planning horizon is selected to 1.5 seconds, with an input noise covariance matrix defined as $\sigma_{\boldsymbol{u}} = \mathrm{diag}(0.2, 0.2)$. The sample size for the MPPI was chosen to be $N_{sample} = 2000$. The MPPI costs in (4.31) were selected to be $Q = \mathrm{diag}(50, 50)$, $R = \mathrm{diag}(1, 1)$, $Q_f = \mathrm{diag}(200, 200)$, and $\alpha_o = \alpha_{ISS} = 10000$. The MPPI inverse temperature parameter λ is chosen to be 0.1. For the cost map, $\alpha_{shift} = 0.1$, $w_{map} = l_{map} = 200$, and $r_{map} = 0.05$ m, with the map being stationary and centered at the position $(0, 0)$.

Controller Specifics

The effectiveness of the controller, as outlined in Theorem 3, is contingent on the existence of the parameter, $\tilde{\lambda}_1$. However, the time derivative of the Lyapunov function in (4.8) does not depend on the polar error state δ, which implies that $\tilde{\lambda}_1$'s existence cannot be assured. Focusing on the primary objective of tracking the desired positions $\boldsymbol{p}^d(t)$, the reduced polar coordinate error dynamics are considered as follows:

$$\dot{\hat{\boldsymbol{e}}} = \frac{d}{dt}\begin{bmatrix}\rho \\ \gamma\end{bmatrix} = \begin{bmatrix}-\cos(\gamma) & 0 \\ \frac{\sin(\gamma)}{\rho} & -1\end{bmatrix}\delta\boldsymbol{u} = \hat{g}_p(\hat{\boldsymbol{e}})\delta\boldsymbol{u}. \tag{4.40}$$

Specifically, the input of the above dynamics is now $\boldsymbol{u} = \hat{\kappa}(\hat{\boldsymbol{e}}) = [\hat{v}, \hat{\omega}]^T$, where $\hat{v} = k_1\rho\cos\gamma$ and $\hat{\omega} = k_2\gamma + k_1\sin(\gamma)\cos(\gamma)$. With the Lyapunov function $\hat{V} = \frac{1}{2}(\rho^2 + \gamma^2)$, the controlled system (4.40) is exponentially stabilizing to $(0, 0)$ within domain $\mathcal{D}^\rho \setminus \mathcal{D}^{dz} \times \mathcal{D}^\gamma$ with $\dot{\hat{V}} \leq -\alpha_3\|\hat{\boldsymbol{e}}\|$ and $\alpha_3 = \frac{k_2\rho_{dz}}{2\sqrt{(\pi/2)^2 + \rho_{dz}}}$. Choosing $k_1 = 0.3$ and $k_2 = 0.15$, the perturbed closed-loop system has a Lipschitz constant $l_{\hat{V}} = \pi/2$, which is obtained numerically over an input domain. For tracking purposes, the convergence of $\hat{\boldsymbol{e}}$ to zero implies achieving the desired output tracking, *i.e.*, $\boldsymbol{p} \longrightarrow \boldsymbol{p}^*$.

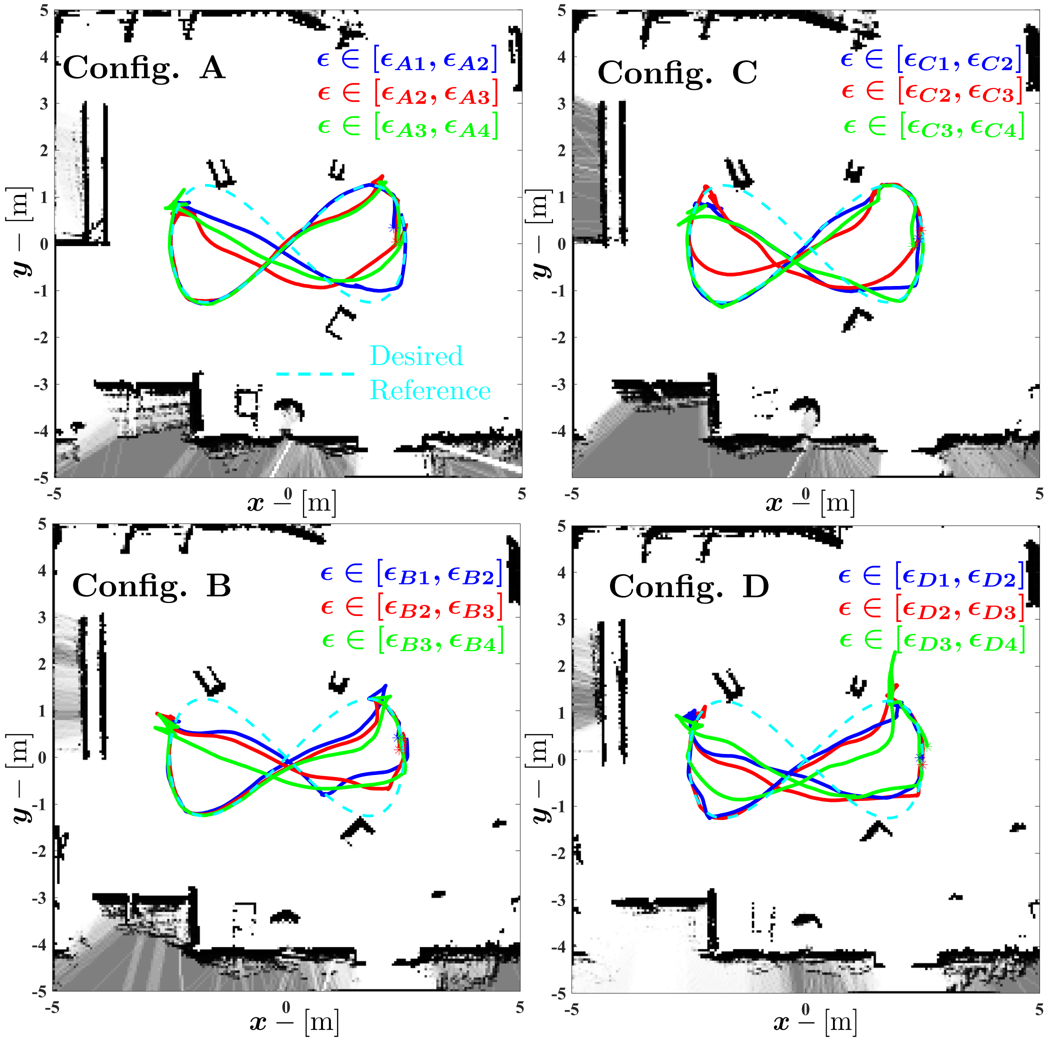

Figure 4.8: Experimental results of 30-second laps “Figure 8” tracking in the presence of stationary obstacles with test configurations A, B, C, and D. The black pixels in the backgrounds for each configuration are the obstacle-occupied grids from LiDAR measurements. The obstacles are placed to obstruct the vehicle if not avoided. The unsafe probability ϵ ranges for the four configurations are $\epsilon_{A1} = 0.1760$, $\epsilon_{A2} = 0.0029$, $\epsilon_{A3} = 0.0010$, $\epsilon_{A4} = 0.0005$, $\epsilon_{B1} = 0.1409$, $\epsilon_{B2} = 0.0016$, $\epsilon_{B3} = 0.0013$, $\epsilon_{B4} = 0.0011$, $\epsilon_{C1} = 0.1321$, $\epsilon_{C2} = 0.0024$, $\epsilon_{C3} = 0.0011$, $\epsilon_{C4} = 0.0009$, $\epsilon_{D1} = 0.0031$, $\epsilon_{D2} = 0.0016$, $\epsilon_{D3} = 0.0012$, and $\epsilon_{D1} = 0.001$.

Config. | ϵ	Param.	0.001	0.005	0.01
A	$Z_\epsilon \mid Z_\epsilon^\perp$	0.710 | 0.448	0.423 | 0.025	0.393 | 0.019
	$r_0 \mid r_{\Delta T}$	0.252 | 0.255	0.090 | 0.090	0.077 | 0.077
B	$Z_\epsilon \mid Z_\epsilon^\perp$	2.153 | 0.034	0.419 | 0.030	0.381 | 0.026
	$r_0 \mid r_{\Delta T}$	2.318 | 2.320	0.088 | 0.088	0.073 | 0.073
C	$Z_\epsilon \mid Z_\epsilon^\perp$	1.113 | 0.032	0.413 | 0.027	0.369 | 0.020
	$r_0 \mid r_{\Delta T}$	0.619 | 0.620	0.085 | 0.085	0.068 | 0.068
D	$Z_\epsilon \mid Z_\epsilon^\perp$	1.878 | 0.095	0.429 | 0.032	0.401 | 0.031
	$r_0 \mid r_{\Delta T}$	1.763 | 1.768	0.092 | 0.092	0.080 | 0.081

Table 4.1: Summary of the offline conformal discrepancy training results and the augmented controller tracking guarantees where autonomous trajectory tracking is performed without obstacles in the four configurations. The training $1-\epsilon$ confidence upper bounds for $\|\boldsymbol{d_u}\|$ and $|dd_\perp|$ is provided at three ϵ levels using Algorithm 2. Under the augmented control policy κ_{ISS}, the maximum tracking error tube radii r_0 and $r_{\Delta T}$ with the choices of ϵ levels are tabulated. Note, the identified radii r_0, $r_{\Delta t}$ are in meters.

Applying Theorem 3 to the reduced polar coordinate error dynamics in (4.40), and with the parameter choices, $\alpha_1 = \alpha_2 = 0.5$ is chosen. The class K_∞ function $\alpha_3(V) = 0.0024V$ is affine. Most importantly, there exists $\hat{\lambda}_1 = 1000$ which leads to $\tilde{\alpha}_3(V) = -0.0038V$. With the augmented ancillary controller $\hat{\kappa}_{ISS} = \hat{\kappa}(\hat{\boldsymbol{e}}) - \frac{1}{\hat{\lambda}_1}\hat{g}_p(\hat{\boldsymbol{e}})^T\hat{\boldsymbol{e}}$, the radii r_0 and $r_{\Delta T}$ are computed for the 4 configurations consolidated in Table 5.1. The overall policy that is sent to the vehicle is:

$$\boldsymbol{u}_{cmd} = \min(\max(\boldsymbol{u}_0^* + \hat{\kappa}_{ISS}, \boldsymbol{u_{min}}), \boldsymbol{u}_{max}). \tag{4.41}$$

Results

The proposed discrepancy-aware MPPI is validated by tracking the same "Figure 8" trajectory used for collecting training data, denoted as $\boldsymbol{p}^d = (x^d(t), y^d(t))$, with a lap time of $T_{lap} = 30$ sec [8]. The experimental setup can be found in Fig. 4.5. As highlighted in the supplementary video [163], the nominal MPPI planner combined with the nominal controller without accounting model mismatches, failed to ensure safety, resulting in a collision with the stationary crates. In contrast, the proposed framework in this chapter completed the trajectory tracking tasks across all four vehicle-ground configurations, effectively handling model mismatches while providing a verifiable safe traversal confidence level. The corresponding hardware

[8] Despite the validation desired trajectory's geometry matches with the training set, the vehicle must deviate from the reference trajectory to avoid the stationary obstacles, covering untrained positions

results are detailed in Fig. 4.8 given the four vehicle-ground configurations, see Fig. 4.2.

A key feature of the proposed approach is the augmentation of the cost map on a grid basis. By associating each increment in N_ϵ with a specific range of ϵ values, the framework exhibits risk-aware tunability. It was observed that lower values of ϵ (approaching zero) induce a conservative obstacle-avoiding behavior, akin to traditional robust control methods. Conversely, as ϵ nears one, the risk-neutral case, the vehicle tracks closely with the reference path, shows improvements in tracking performance at the cost of safety confidence.

During the post-experiment analysis, particularly with configurations B and D, Large model discrepancies for $\epsilon = 0.001$ are observed. These discrepancies were predominantly found in datasets characterized by lower lap speeds, suggesting a correlation with specific state or input conditions. This observation signals the possibilities of a state and input-dependent discrepancy [13], which could potentially provide a more accurate safety buffer compared to the upper bounds Z_ϵ and $Z_\epsilon^\perp$.

4.6 Hardware Validation: Driver-Assist with Collision Avoidance Human-in-the-Loop Setup

In this section, the application of the proposed framework in collision driver assist scenarios is explored. Consider a driver remotely operating a UGV robot via a joystick, with linear velocity command in the body frame's x axis (v_{joy}) and angular velocity command in the z axis (ω_{joy}), same as the input commands in (4.1). The driver assist program aims to follow the joystick commands while providing avoiding obstacles up to the specified risk tolerance.

Drawing from Hugemann *et. al.* [164]'s study on driver reaction times in road traffic, a conservative reaction time of 1.5 seconds is considered for collision avoidance. This 1.5 seconds can be interpreted as the reaction time as a 1.5 second ZOH to the most recent joystick commands. Specifically, the desired reference generated using the joystick trajectory $\{\boldsymbol{x}_{t+k}\}_{k=1}^{n_h}$, can be computed as

$$\boldsymbol{x}_{t+k}^{joy} = \boldsymbol{x}_t + \sum_{i=0}^{k} g(\boldsymbol{x}_{t+i}) \begin{bmatrix} v_{joy} \\ \omega_{joy} \end{bmatrix} \Delta T, \tag{4.42}$$

where $k \in \mathbb{Z}_1^{n_h}$. To ensure maximum teleportation control in collision-free environments, the proposed framework is modified to only provide trajectory correction when a collision is imminent based on the joystick trajectory.

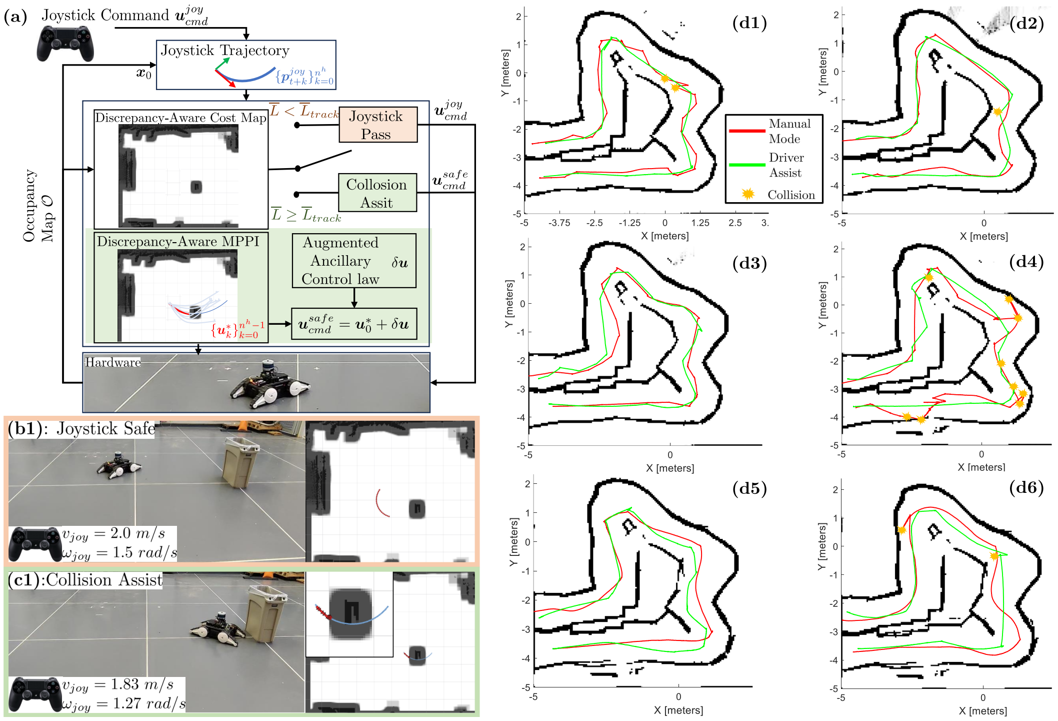
(a) Joystick Command u_{cmd}^{joy}
Joystick Trajectory
x_0
$\{p_{t+k}^{joy}\}_{k=0}^{n^h}$
Discrepancy-Aware Cost Map
$\bar{L} < \bar{L}_{track}$
Joystick Pass
u_{cmd}^{joy}
Collision Assit
u_{cmd}^{safe}
$\bar{L} \geq \bar{L}_{track}$
Occupancy Map $\mathcal{O}$
Discrepancy-Aware MPPI
Augmented Ancillary Control law δu
$\{u_k^*\}_{k=0}^{n^h-1}$
$u_{cmd}^{safe} = u_0^* + \delta u$
Hardware
(b1): Joystick Safe
$v_{joy} = 2.0\ m/s$
$\omega_{joy} = 1.5\ rad/s$
(c1):Collision Assist
$v_{joy} = 1.83\ m/s$
$\omega_{joy} = 1.27\ rad/s$
(d1)
(d2)
(d3)
(d4)
(d5)
(d6)
Manual Mode
Driver Assist
Collision
X [meters]
Y [meters]
-5
-3.75
-2.5
-1.25
0
1.25
2.5
3.
2
1
0
-1
-2
-3
-4
-5

Figure 4.9: Experimental results of using the proposed framework performing collision avoidance assistance for human drivers. Based on human-provided joystick commands and a user-chosen risk tolerance ϵ, the collision risk of joystick trajectory based on current state feedback is evaluated on the proposed discrepancy-aware cost map. If the collision is projected based on the evaluated cost, the proposed planning and control framework will activate to provide overriding safe commands. Sub-figure (a) is a diagram that summarizes the driver-assist logic flows. Sub-figures (b1) and (b2) illustrate the scenario for safe joystick inputs where no collision assist is inactive. Sub-figures (c1) and (c2) illustrate the scenario where the projected joystick trajectory is unsafe and where the collision assist program is active to provide optimal and safe commands. Sub-figures (d1)-(d6) are the six test subjects' trajectories in manual mode and driver assist mode.

In this experiment, human subjects drive the UGV in configuration A without a direct view of the test course. The test drivers rely solely on a 2D occupancy map for navigating among obstacles, matching the perception capability of the framework. The human drivers are tasked to finish the narrow test course without collision as quickly as possible.

Note, the same training set presented in section 4.5 is used in the driver assist application despite the vehicle trajectory does not match the training Figure 8 trajectory to highlight the learned upper bounds Z_ϵ and $Z_\epsilon^\perp$ is training trajectory independent.

Driver-Assist Implementation

Using the training result tabulated in table 5.1, the discrepancy-aware cost map is constructed with a risk value ϵ ranges in $(0.0029, 0.0010)$. the joystick trajectory against the cost map is first evaluated to determine a joystick trajectory cost:

$$\overline{\mathcal{L}}_{joy} = \sum_{k=1}^{n_h} \alpha_{joy} \mathcal{L}_C(\boldsymbol{p}_{t+k}^{joy}, C_\epsilon), \tag{4.43}$$

where $\alpha_{joy} \in \mathbb{R}_{>0}$ is an adjustable parameter default to be 1.0. The cost-to-go and input cost are set to zero, considering the joystick trajectory itself is the desired reference. If $\overline{\mathcal{L}}_{joy}$ is less than $\overline{\mathcal{L}}_{track}$, the joystick trajectory is deemed safe with a confidence level of $1 - \epsilon$ for the immediate 1.5 seconds. Conversely, if $\overline{\mathcal{L}}_{joy} \geq \overline{\mathcal{L}}_{track}$, there is a potential collision, requiring a safety override.

The proposed framework is shown to be capable of producing such safety override. The MPPI algorithm is initialized with the joystick input as a warm start, and the MPPI input perturbation covariance set as a diagonal matrix of 0.25 m/s and $0.25 rad/s$ for linear and angular velocities. Since the maximum joystick linear and angular speeds are 2 m/s and 2 rad/s, a zero-speed (stopping) command cannot be guaranteed to be sampled. As a solution, the turn-in-place (TIP) motion primitives are synthesized and added to the flow sampling and input aggregation in addition to the joystick trajectory. Specifically, the joystick turning command with 0 linear velocity as $\{\boldsymbol{u}_k^{TIP}\}_{k=0}^{k=n_h-1} = [0, \omega_{joy}]^T$ are used which allow the driver to perform heading adjustment freely but not the vehicle positions because of potential collision risks in the current heading direction.

With a total MPPI sample size of $N_{sample} = 5000$ and inverse temperature $\lambda = 0.05$, 80% of sampled input discrepancies are allocated to the joystick input sequences, and the remaining samples are aggregated on top of the TIP motion primitives. Following Algorithm (3) to compute the associated costs of each sampled flow, the collision-assist safety override $\boldsymbol{u}_{cmd}^{safe}$ is the MPPI solution. [9] A detailed diagram of the collision-assist program is presented in Fig. (4.9).

The user-selected parameter α_{joy} is set to be $\alpha_{joy} = 1/k^2$, inversely proportional to the prediction horizon squared. As the collision-assistance program recalculates at 20 Hz, this cost decay over future horizon steps is chosen to reduce planner conservatism.

[9] It is always desirable to use the joystick trajectory as a warm start without using the previous MPPI solution. Such engineering decision is made to ensure maximum adherence to the driver's command which can be discontinuous and far from the previous projected joystick trajectory.

Subj.	Seq.	# C (M)	T_t (M)	# C (CA)	T_t (CA)
1	M/DA	2	63.7	0	38.8
2	M/DA	0	50.7	1	40.3
3	DA/M	0	41.5	0	44.8
4	M/DA	7	107.5	3	63.6
5	DA/M	0	45.6	0	30.1
6	DA/M	1	35.6	1	28.3

Table 4.2: Summary of the diver assist program with 6 different human drivers. Each subject is assigned to attempt to complete the task with either manual mode (M) or the collision assistance mode (CA). The mode experience order is indicated in the second column. The number of collisions ($\#C$) during the test drives is recorded and the test drive duration T_t.

Results and Discussion

An experimental comparison with six human drivers navigating a narrow pathway is conducted, with various experience levels for driving the vehicle remotely. Each test subject will drive the same test course twice, once with and once without the proposed collision assistive program. The order of the testing is randomized to minimize any bias due to the driver's familiarity with the test course. The test results are tabulated in Table 4.2, with comparative drive trajectories depicted in Fig. 4.9.

Based on the tabulated results, it is observed that the proposed collision-assistive program showed minor improvements for rather inexperienced drivers (such as drivers 1 and 4) who faced challenges during their initial manual run. Despite the driving sequence being assigned randomly, a consistent reduction in driving time was observed, likely attributable to the gradual accumulation of track and vehicle knowledge. Notably, subjects who initially experienced manual driving followed by the driver-assist program exhibited percent drive time improvements of 39.1%, 20.5%, and 40.8%. Conversely, subjects who first used the assistive program and then switched to manual mode showed improvements of −8.0%, 34.0%, and 20.5%. This variance in performance improvement may be attributed to enhanced driver confidence in collision avoidance due to the assistive program.

The feedback from test subjects on the driver-assist program was mixed. Positive remarks primarily center around the program's effectiveness and the ability to provide collision-avoiding heading adjustments automatically. However, several drivers noted instances where the vehicle, under driver-assist control, executed minor reverse maneuvers despite receiving forward velocity commands being counterintuitive. This behavior is the result of the model mismatches, where the nominal model predicts a trajectory leading into high-risk or collision-prone areas, prompting the program to revert to a safer region. To improve user experience and program interpretation, the development of a more intuitive driver interface is recommended. Furthermore, integrating user-preference-based cost tuning could better tailor individual drivers' preferences in balancing between reference tracking and collision safety, as suggested in recent studies such as [165], [166]. It is important to note that due to the limited number of participants, there will be no statistically significant conclusions drawn from this study.

4.7 Results and Discussion

In summary, this chapter provides a novel multi-layered framework designed to provide safety-critical autonomy in the presence of obstacles and model discrepancies. The framework's core involves data-driven discrepancy identification, extracting both matched and unmatched model residuals from offline data with minimal assumptions. These identified discrepancies are then used to augment the vehicle's ancillary controller, offering stability assurances for the closed-loop system. The framework is completed with a discrepancy-aware MPPI planner that generates (sub)optimal and safe reference tracking paths, taking into account imperfection in actual trajectory tracking due to model discrepancies.

The proposed framework is theoretically supported throughout its construction. By deducing the maximum tracking error resulting from matched and unmatched model discrepancies, safety and robustness are rigorously ensured by assessing the interactions between the planner and controller layers. The proposed framework is validated through extensive hardware experiments, demonstrating its effectiveness in trajectory tracking in cluttered environments. Additionally, the framework is successfully adapted to a driver-assist program, providing optimal, safe assistive commands in potential collision scenarios.

Limitation and Future Works

The current limitations of the proposed framework include potential safety violations in importance sampling aggregation law, a tendency towards planning overly conservative trajectories, and vulnerability to training outliers in model discrepancies. Based on the choice of the inverse temperature λ, $\lambda \to \infty$ plans smoother trajectories with equal weighting, and $\lambda \to 0$ equates to sample rejection that could lead to instability [113]. However, sampled input aggregation as $\lambda \to \infty$ could lead to an unsafe aggregated trajectory where safe and unsafe trajectories are less distinguished. Nevertheless, $\lambda \to 0$ prioritizes picking safe trajectories but may lead to overly conservative and chattered trajectories from randomly sampled input sequences.

On the discrepancy identification front, conformal prediction relies on empirical distribution quantiles instead of concentration inequalities allowing it to be more sample-efficient. However, in the presence of large samples, potential outliers, and small risk tolerance, the identified Z_ϵ and $Z_\epsilon^\perp$ can be falsely large yielding overly conservative results. Sufficient data set prepossessing might also be required to remove outliers.

Looking ahead, future work aims to refine the offline conformal-driven discrepancy upper bound component into an online algorithm. Adaptive conformal prediction methods, akin to the one detailed by [14], to actively detect distribution shifts from training sets due to terrain or environmental changes, exhibit promising potential. Moreover, a higher fidelity nominal model such as a learning-based model can be used to enhance tracking performance. Another future research involves the theoretical analysis of the discrepancy-aware MPPI in terms of optimality and recursive feasibility. The authors are also investigating the possibilities rejecting of unsafe MPPI trajectory samples, incorporating motion primitives, and incorporating generalized state constraints beyond obstacle avoidance constraints.

Chapter 5

MOA: LINEAR RECURRENCE AND BOOTSTRAPPING

This chapter presents a new framework for discovering the dynamics of *a priori* unknown moving obstacles, forecasting their trajectories, and providing risk-aware optimal avoidance strategies. The proposed framework replaces the need for obstacle trajectory/model classification while allowing online computation. Extracting a dynamics model from data is challenging [65], especially when the available data is limited, noisy, and partial. Takens' embedding theorem [66] is a technique for tackling partial measurements, which uses partial observations to produce an attractor that is diffeomorphic to the full-state attractor. Singular Spectrum Analysis (SSA) [87], [88] is a time series technique for separating noise from the underlying signal and extracting a predictive model of obstacle behavior. Inspired by [89], set obstacle trajectories forecast is made by bootstrapping obstacle models extracted using obstacle measurements from different time snapshots.

An MPC planner then incorporates the set of obstacle forecasts as an affine conservative approximation of a distributionally robust chance constraint (DRCC). This constraint is then efficiently recast in a risk-aware manner, allowing an MPC optimization based on sequential convex programming [90], [91].

This chapter was adapted from: [15]

5.1 Problem Statement

Consider the linear, discrete-time ego vehicle dynamics model:

$$\boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\boldsymbol{u}_t, \qquad \boldsymbol{y}_{t+1} = G\boldsymbol{x}_{t+1} \tag{5.1}$$

where $\boldsymbol{x}_t \in \mathcal{D}^{\boldsymbol{x}} \subseteq \mathbb{R}^{n_x}$, $\boldsymbol{u}_t \in \mathcal{D}^{\boldsymbol{u}} \subseteq \mathbb{R}^{n_u}$, and $\boldsymbol{y}_t \in \mathbb{R}^{n_y}$, for all $t \in \mathbb{N}$, correspond to the system states, controls, and outputs at time index t, respectively. The state transition, actuation, and measurement matrices are $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, and $G \in \mathbb{R}^{n_y \times n_x}$, respectively. Constant matrix $C \in \mathbb{R}^{3 \times n_x}$ maps the system's states (5.1) to the system's x, y, z positions with respect to inertial frame E. The k^{th} obstacle, $k \in \mathbb{Z}^{1:N^{\text{obs}}}$, is modeled as a sphere. The obstacle occupies the point set

$$O_k(\boldsymbol{c}_k, r_k) = \{\boldsymbol{x} \in \mathbb{R}^3 : \|\boldsymbol{c}_k - \boldsymbol{x}\|_2 \leq r_k\},$$

where $\boldsymbol{c}_k \in \mathbb{R}^3$ and $r_k \in \mathbb{R}_+$ are the k^{th} obstacle's center and radius.

Considering the case where the agent (5.1) is tasked with following a reference output trajectory $\boldsymbol{y}^{\text{ref}}$ which need not consider obstacle information. While following this path, the agent may encounter N^{obs} spherical, stationary or moving obstacles. The obstacle-free region is the open set:

$$\mathcal{S} \triangleq \left\{ \mathbb{R}^3 \setminus \cup_{k=1}^{N^{\text{obs}}} \mathcal{O}_k \right\}. \tag{5.2}$$

Assumption 4. *Obstacles can be detected and localized at the same rate (f^+ Hz) as the planner update. Only measurements of an obstacle's geometric center with respect to frame E are assumed, and they are corrupted by a zero-mean noise. Suppose a radius, r_k, of the k^{th} obstacle as $\hat{r}_k$, can be estimated which satisfies $\hat{r}_k \geq r_k$. Note, Assumption 4* ***does not*** *imply full state measurement.*

Assumption 5. *All obstacle measurements, admit an L-decomposition of order d, are governed by LRFs* (5.5) *whose LRF coefficients can be uniquely defined.*

Assumption 6. *The obstacles' velocities are assumed to be bounded by v_{max}, and the initial distances between all obstacles and the agent are significantly greater than $\frac{dv_{max}}{f^+}$.*

Problem 2. ***[Prediction]*** *Consider a multivariate stochastic process where observables $\{x\}_1^N$, $\{y\}_1^N$, and $\{z\}_1^N$ denote the spherical obstacle's true center location in reference frame, E.*

The measurements are corrupted by independent, zero-mean noises $\{\gamma_1\}_1^N$, $\{\gamma_2\}_1^N$, and $\{\gamma_3\}_1^N$ (see Fig. 5.1). Under Assumptions 4-6, the problem objective is to predict the obstacle position at times $N+1$ to $N+n_h$ using these measurements where $n_h \in \mathbb{Z}_{>0}$.

Due to limited and noisy partial data and the lack of explicit dynamics models,

Preserving the stochastic nature of the problem, it is desirable to estimate a Bootstrap distribution of the obstacle predictions, denoted by the random set O^{pred}, from time index $N+1$ to $N+n_h$ and calculate its first and second moments. More specifically, the discrepancies in the forecast due to inadequate signal and noise separation and bandwidth limits (both for the case of not enough training data and incorrect window length L for embedding) are expected. This leads to the following chance constraint model predictive planning problem formulation, the DRCC MPC problem.

Problem 3. ***[Planning]*** *Consider the system* (5.1) *and free-space* (5.2)*. Given a discrete-time reference trajectory* $\boldsymbol{y}^{ref}_{t+i}\,\forall i \in \mathbb{Z}^{n_h}_1$ *where* $n^h \in \mathbb{Z}_{>0}$ *is the length of the horizon, convex state constraints* $\mathcal{D}^{\boldsymbol{x}} \subseteq \mathbb{R}^{n_{\boldsymbol{x}}}$*, convex input constraints* $\mathcal{D}^{\boldsymbol{u}} \subseteq \mathbb{R}^{n_{\boldsymbol{u}}}$*, and a convex stage cost* $\mathcal{L}_i : \mathbb{R}^{n_{\boldsymbol{x}}} \times \mathbb{R}^{n_{\boldsymbol{u}}} \rightarrow \mathbb{R}_{\geq 0}$*, a total of* N^{obs} *spherical obstacles each approximated by a set* O^{pred}_k*, and risk tolerance* $\epsilon \in (0, 1]$*, the problem objective is to compute a receding horizon controller* $\{\boldsymbol{u}^*_{t+i}\}^{n_h-1}_{i=0}$ *that avoids the unsafe set* $O^{pred} \triangleq \bigcup^{N^{obs}}_{k=1} O^{pred}_k$ *via the following non-convex optimization:*

$$\{\boldsymbol{u}^*_{t+i}\}^{n_h-1}_{i=0} = \min_{\{\boldsymbol{u}_i\}^{n_h-1}_{i=0} \in \mathbb{R}^{n_{\boldsymbol{u}}}} \sum^{n_h}_{i=0} \mathcal{L}_i(\boldsymbol{y}^{ref}_{t+i+1} - \boldsymbol{y}_{t+i+1}, \boldsymbol{u}_i) \tag{5.3a}$$

$$s.t. \quad \boldsymbol{x}_{i+1} = A\boldsymbol{x}_i + B\boldsymbol{u}_i \quad \boldsymbol{y}_{t+i+1} = G\boldsymbol{x}_{i+1} \tag{5.3b}$$

$$\boldsymbol{x}_{i+1} \in \mathcal{D}^{\boldsymbol{x}}, \quad \boldsymbol{u}_i \in \mathcal{D}^{\boldsymbol{u}}, \quad \boldsymbol{x}_0 = \boldsymbol{x}_{t,init} \tag{5.3c}$$

$$\mathbb{P}(\boldsymbol{x}_i \in O^{pred}) \leq \epsilon, \quad \forall i \in \mathbb{Z}^{n_h-1}_0. \tag{5.3d}$$

5.2 Singular Spectrum Analysis

Consider a discrete-time multivariate stochastic process $\{\boldsymbol{o}^m_t\}^N_{t=1}$ where m denotes the m^{th} observable measurement of the process, and N is the total number of available observations. *i.e.*, $\boldsymbol{o}^m_t$ denotes the m^{th} observation variable at process sampling index t. Suppose that the true stochastic process model of the observables is $\hat{\boldsymbol{o}}^m_t = \boldsymbol{o}^m_t + \boldsymbol{\gamma}^m_t$, where $\boldsymbol{\gamma}^m_t$ denotes a random discrete-time zero-mean measurement noise, and $\boldsymbol{o}^m_t$ is the noiseless observable that captures the governing laws, which can be composed of *trends*, *seasons*, and *stationary* time series. Singular Spectrum Analysis [87] separates the true signal $\boldsymbol{o}^m$ and the noise $\boldsymbol{\gamma}^m$ and extracts a recursive governing dynamic model of $\boldsymbol{o}^m$ that can generate a short term accurate forecast. Fig. 5.1 describes this method.

Time Delay Embedding

Takens' method of delays, introduced by [66], can reconstruct qualitative features of the full-state phase-space from delayed partial observations. The m^{th}-state observables $\hat{\boldsymbol{o}}^m_t$ are delay embedded into a trajectory (Hankel) matrix $H^{m,L,N}_t$, Fig. 5.1 gives an example of the Hankel matrix for state x. Parameter L is the time delay length, and N is the time series length. Repeating patterns in the Hankel matrix represent underlying trends and oscillations, which can be extracted from its covariance matrix:

$$X^m_t = H^{m,L,N}_t (H^{m,L,N}_t)^T.$$

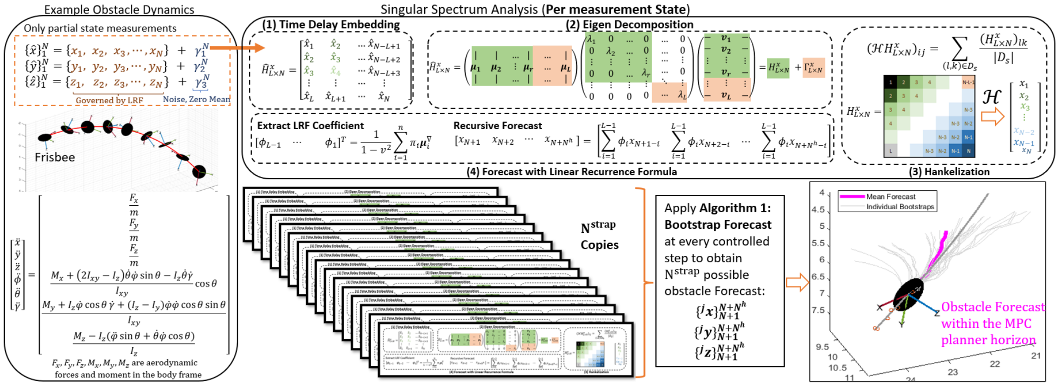
Example Obstacle Dynamics
Only partial state measurements
Governed by LRF
Noise, Zero Mean
Frisbee
$F_x, F_y, F_z, M_x, M_y, M_z$ are aerodynamic forces and moment in the body frame
Singular Spectrum Analysis (Per measurement State)
(1) Time Delay Embedding
(2) Eigen Decomposition
(3) Hankelization
(4) Forecast with Linear Recurrence Formula
Extract LRF Coefficient
Recursive Forecast
N^strap Copies
Apply Algorithm 1: Bootstrap Forecast at every controlled step to obtain N^strap possible obstacle Forecast:
Mean Forecast
Individual Bootstraps
Obstacle Forecast within the MPC planner horizon

Figure 5.1: A description of bootstrap-SSA-forecast architecture in forecasting the trajectory of a Frisbee where the stochastic observables (corrupted by zero-mean, noise) consist of $\{\hat{\boldsymbol{o}}_t\}_{t=1}^N = [\{\hat{x}_t\}_{t=1}^N, \{\hat{y}_t\}_{t=1}^N, \{\hat{z}_t\}_{t=1}^N]$, the Frisbee's center positions with respect to an inertial frame. The SSA analysis and bootstrap forecast are applied to every observable state. Despite its 12-state governing dynamics [167] and with only center position measurements of the Frisbee, an example N^{strap} forecasts of the Frisbee trajectory for future time steps $\{1, 2, \cdots, n_h\}$ is depicted using the proposed framework.

More details regarding time-delay embedding and the construction of the Hankel matrix can be found in Chapter 2.

Eigen Decomposition

To recover the true signal $\boldsymbol{o}^m$, a low-rank matrix approximation of this signal by thresholding the eigenvalues of X^m [168] is desirable. The symmetric covariance matrix X^m has a spectral decomposition $U\Sigma U^T$, where Σ is a diagonal matrix with real eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_L$. The matrix of left eigenvectors $U = \begin{bmatrix} \boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_L \end{bmatrix}$ is orthogonal. The truncated right eigenvectors $V = [\boldsymbol{\nu}_1, \cdots, \boldsymbol{\nu}_L]^T \in \mathbb{R}^{L\times N}$ of X^m can be found as $V = U\Sigma$. Suppose λ^* is the optimal threshold for separating the signal $\boldsymbol{y}^m$ from the noise $\boldsymbol{\gamma}^m$, and $\lambda_n \geq \lambda^* \geq \lambda_{n+1}$, which partitions the Hankel matrix $H_t^{m,L,N}$ as:

$$H_t^{m,L,N} = \underbrace{\sum_{p=1}^{n} \sqrt{\lambda_p}\boldsymbol{\mu}_p\boldsymbol{\nu}_p^T}_{\triangleq H_t^{o,L,N}} + \underbrace{\sum_{p=n+1}^{L} \sqrt{\lambda_p}\boldsymbol{\mu}_p\boldsymbol{\nu}_p^T}_{\triangleq H_t^{\gamma,L,N}} . \tag{5.4}$$

Unlike standard SSA [87], which aims to separate a time series' trends and periodic structures, noiseless obstacle's behavior is required which allows us to extract its governing dynamics. To account for incomplete noise and signal separation and low data limits, a technique of progressively relaxing singular value threshold and sample bootstrapping is proposed in Section 5.3.

Hankelization

Matrix $H_t^{o,L,N}$ in (5.4) should maintain a Hankel structure: minor variations in its k^{th} secondary diagonals result from insufficient noise removal. The k^{th} secondary diagonals of a matrix M are also the k^{th} diagonals of M flipped horizontally with respect to its middle column. A *Hankelization step* performs secondary diagonal averaging in order to find the matrix $\mathcal{H}O$ that is closest to $H_t^{o,L,N}$ with respect to the Frobenius norm among all $L \times N$ Hankel matrices [87]. The operator $\mathcal{H}$ acting on $L \times N$ matrix $H_t^{y,L,N}$ entry wise is defined as: for the $(i,j)^{th}$ element of matrix $H_t^{o,L,N}$ and $i + j = s$, define a set

$$D_s \triangleq \{(l,n) : l + n = s, l \in \mathbb{Z}_1^L, n \in \mathbb{Z}_1^N\},$$

is mapped to $(i,j)^{\text{th}}$ element of the Hankelized $\mathcal{H}H_t^{o,L,N}$ via the expression in Fig.5.1 (for the case of $\boldsymbol{o}^m = \boldsymbol{x}$), where $|D_s|$ denotes the number of elements in set D_s.

Forecast with Linear Recurrence Formula

Definition 9. *A time series $Y_N = \{y\}_1^N$ admits an L-decomposition of order not larger than d, denoted by $ord_L(Y_N) \leq d$, if there exist two systems of functions $\varrho_k : \mathbb{Z}_0^{L-1} \to \mathbb{R}, \vartheta_k : \mathbb{Z}_0^{L-1} \to \mathbb{R},$ such that*

$$y_{i+j} = \sum_{k=1}^{d} \vartheta_k(i)\varrho_k(j), \quad \{i,j\} \in \mathbb{Z}_0^{L-1} \times \mathbb{Z}_0^{L-1}$$

$\forall k \in \mathbb{Z}_1^d.$

If $\text{ord}_L(Y_N) = d$, then the series Y_N admits a *L-decomposition of the order d* and both systems of functions $(\varrho_1, \cdots, \varrho_d)$ and $(\vartheta_1, \cdots, \vartheta_d)$ are linearly independent [169].

Definition 10. *A time series $\{y\}_1^N$ is governed by a linear recurrent relations/formula (LRF), if there exist coefficients $\{\phi\}_1^d$ and $\phi_d \neq 0$ such that*

$$y_{i+d} = \sum_{k=1}^{d} \phi_k y_{i+d-k}, \quad \forall i \in \mathbb{Z}_0^{N-d}, d < N \ . \tag{5.5}$$

Real-valued time series governed by LRFs consists of sums of products of polynomials, exponentials and sinusoids [87].

Theorem 5. *[87] Let $\boldsymbol{\mu}_i^{1:L-1}$ be the vector of the first $L-1$ components of a left eigenvector $\boldsymbol{\mu}_i$ of $H_t^{m,L,N}$, and let π_i be the L^{th} component of eigenvector $\boldsymbol{\mu}_i$. Let $v^2 \triangleq \sum_{i=1}^{d} \pi_i^2$. Under Assumptions 5 and 6 (see below), the LRF coefficients ϕ_i where $i \in [1, L-1]$ can be computed as:*

$$\begin{bmatrix} \phi_{L-1} & \phi_{L-2} & \cdots & \phi_1 \end{bmatrix}^T = \frac{1}{1-v^2} \sum_{i=1}^{d} \pi_i \boldsymbol{\mu}_i^{1:L-1}, \tag{5.6}$$

and $\boldsymbol{y}$ evolves as the LRF:

$$\boldsymbol{y}_{N+1} = \sum_{j=1}^{L-1} \phi_j \boldsymbol{y}_{N-j}.$$

5.3 Bootstrap Forecasting

Despite empirical successes in reconstructing and forecasting [88], the theoretical accuracy of SSA is strenuous to obtain, see [170]. Inspired by [89], bootstrapping can be used to improve model discovery and to produce probabilistic forecasts.

The real-time bootstrap forecast, Algorithm 4, assumes time series measurements corrupted by noise. The user-defined parameters N^{train} and N^{step} represent the number of initial training samples, and the number of newly accumulated samples during an initial bootstrap. Further, one must choose parameters δ_t and N^{σ} , where threshold δ_t is used to separate signal from noise, and N^{σ} is the number of steps of progressive relaxation of threshold δ_t. The parameters δ_t and N^{σ} are dictated by measurement noise levels, which can be characterized off-line in a controlled experimental setting. In the desired signal/noise separation (5.4), the unknown theoretical optimal threshold λ^* must be estimated. Let $Y_N^{\lambda_1:\lambda_d}$ be the Hankelization reconstructed $\hat{\boldsymbol{y}}$ with the eigenvalues $\{\lambda\}_1^d$ and their corresponding right and left eigenvectors. Note, if $d > n$ where $\lambda_n \leq \lambda^* \leq \lambda_{n+1}$, then the norm values

$$||Y_N^{\lambda_1:\lambda_{d+t}} - Y_N^{\lambda_1:\lambda_{d+t+1}}||_2 \approx ||Y_N^{\lambda_1:\lambda_{d+t+1}} - Y_N^{\lambda_1:\lambda_{d+t+2}}||_2$$

since they are comprised of the residual measurement noise. the difference between two consecutive reconstructions with δ_t/N needs to be separated, *i.e.* finding the smallest threshold $t \in \mathbb{Z}_{>0}$ such that:

$$||Y_N^{\lambda_1:\lambda_t} - Y_N^{\lambda_1:\lambda_{t+1}}||_2 - ||Y_N^{\lambda_1:\lambda_{t+1}} - Y_N^{\lambda_1:\lambda_{t+2}}||_2 \leq \frac{\delta_t}{N}. \tag{5.7}$$

Since the selection of the threshold δ_t is crucial, an additional parameter N^{σ} is added to ensure no principle components are lost in $Y_N^{\lambda_1:\lambda_d}$ because of bad choice of δ_t, *i.e.* to avoid $d < n$. The next N^{σ} largest eigenvalues is recommended to be included after the first t eigenvalues in the bootstrapping process. Most importantly, the number of bootstraps, N^{strap}, needs to be determined *a priori*, considering the computation capacity, number of obstacles, and the expected noise level The effectiveness of Algorithm 4 depends highly on the time delay length L, the number of training measurements N^{train}, the number of bootstraps N^{strap}, and the MPC horizon length, n_h. It is recommended that the number of measurements used N^{train} be at least $10n_h$ and that $L = \frac{N^{\text{train}}}{4}$. Bootstrap parameters N^{strap} and N^{step} should be as large as allowed by the computing platform.

Algorithm 4: Bootstrap Forecast Algorithms (Per Obstacle)

```
Data: Obstacle center position measurements $\{\hat{\boldsymbol{x}}_i\}_{i=1}^{N}$, $\{\hat{\boldsymbol{y}}_i\}_{i=1}^{N}$, $\{\hat{\boldsymbol{z}}_i\}_{i=1}^{N}$,
      User-defined constants: $N^{\text{train}}$,$N^{\text{step}}$, $\delta_t$, $N^{\sigma}$, $N^{\text{strap}}$
Result: Forecast:$\{{}^j\boldsymbol{x}_i\}_{i=N+1}^{N+n_h}$, $\{{}^j\boldsymbol{y}_i\}_{i=N+1}^{N+n_h}$, $\{{}^j\boldsymbol{z}_i\}_{i=N+1}^{N+n_h}$, $\forall j \in \mathbb{Z}_1^{N^{\text{straps}}}$
Use $\{\hat{\boldsymbol{x}}_{N+1}, \hat{\boldsymbol{y}}_{N+1}, \hat{\boldsymbol{z}}_{N+1}\}$ to update Hankel matrix
while $istrap \leq N^{strap}$ do
    while $N + 1 \geq N^{train}$ do
        for $states = x, y, z$ do
            while (5.7) holds do
                $t$ ++
            end
            obtain $(\{\lambda_\tau^{\text{istrap}}\}_{\tau=1}^{t}, \{\boldsymbol{\mu}_\tau^{\text{istrap}}\}_{\tau=1}^{t}, \boldsymbol{\phi}^{\text{istrap}})$ for each states
        end
        istrap ++ ;
        for $tt = t + 1 : t + N^{\sigma}$ do
            obtain $(\{\lambda_\tau^{\text{istrap}}\}_{\tau=1}^{tt}, \{\boldsymbol{\mu}_\tau^{\text{istrap}}\}_{\tau=1}^{tt}, \boldsymbol{\phi}^{\text{istrap}})$ for each states, istrap ++
        end
        $N^{\text{train}} = N^{\text{train}} + N^{\text{step}}$
    end
    Back-up Strategy
end
Apply the tuples $(\{{}^j\lambda_\tau^{\text{istrap}}\}_{\tau=1}^{t_j}, \{{}^j\boldsymbol{\mu}_\tau^{\text{istrap}}\}_{\tau=1}^{t_j}, {}^j\boldsymbol{\phi}^{\text{istrap}})$ $\forall j \in \mathbb{Z}_1^{\text{Nstraps}}$ for $x, y, z$ to
 the updated Hankel, where $t_j$ denotes number of eigenvalues post truncation
 for the $j^{th}$ bootstrap. Perform a $n_h$ step forecast using ${}^j\boldsymbol{\phi}^{\text{istrap}}$ to obtain
 $\{{}^j\boldsymbol{x}_i\}_{i=N+1}^{N+n_h}$, $\{{}^j\boldsymbol{y}_i\}_{i=N+1}^{N+n_h}$, $\{{}^j\boldsymbol{z}_i\}_{i=N+1}^{N+n_h}$, $\forall j \in \mathbb{Z}_1^{N^{\text{straps}}}$.
```

5.4 Bootstrap Planning

This section introduces an MPC-based path planner to solve Problem 6. First, the obstacle avoidance constraint (5.3d) is formulated in terms of the mean and variance of the bootstrap predictions. Next, this constraint is incorporated in the MPC optimization to provide probabilistic guarantees given constraint satisfaction.

Algorithm 4 produces N^{strap} copies of n_h length predictions of the k^{th} obstacle's location. Denote the j^{th} copy of the bootstrap prediction as

$$\{\hat{\boldsymbol{y}}_k^j\}_1^{N_h} = \{\hat{\boldsymbol{y}}_{1,k}^j, \hat{\boldsymbol{y}}_{2,k}^j, \cdots, \hat{\boldsymbol{y}}_{n_h,k}^j\}.$$

The collision avoidance set constraint (5.3d) can be reformulated based on the obstacle shape and center as

$$\|C\boldsymbol{x}_i - \hat{\boldsymbol{y}}_{i,k}^j\|_2 \geq \hat{r}_k + r_p \triangleq \overline{r}_k,$$

for each $i \in \mathbb{Z}^{1:n_h}$, $k \in \mathbb{Z}^{1:N^{\text{obs}}}$, and where r_p is the agent's safety radius (5.1). This collision constraint can be reexpressed as the following concave (in the state $\boldsymbol{x}_i$) constraint,

$$(C\boldsymbol{x}_i - \hat{\boldsymbol{y}}_{i,k}^j)^T(C\boldsymbol{x}_i - \hat{\boldsymbol{y}}_{i,k}^j) \geq \overline{r}_k\|(C\boldsymbol{x}_i - \hat{\boldsymbol{y}}_{i,k}^j)\|_2. \tag{5.8}$$

Constraint (5.8) can be approximated as an affine constraint through the use of Sequential Convex Programming (SCP) [90], [91]

$$(C\boldsymbol{x}_i - \hat{\boldsymbol{y}}_{i,k}^j)^T(C\overline{\boldsymbol{x}}_i - \hat{\boldsymbol{y}}_{i,k}^j) \geq \overline{r}_k\|(C\overline{\boldsymbol{x}}_i - \hat{\boldsymbol{y}}_{i,k}^j)\|_2, \tag{5.9}$$

where $\overline{\boldsymbol{x}}_i$ is approximated with the solution from previous SCP iterations. Note that (5.9) over-approximates constraint (5.8) (see [90] for proof).

Lemma 4. *If there are N^{strap} forecasts of the k^{th} obstacle's position from time $i \in \mathbb{Z}^{1:n_h}$ and the previous SCP trajectory $\{\overline{\boldsymbol{x}}\}_1^{n_h}$, the j^{th} bootstrap lumped collision avoidance coefficients $\alpha_{i,k}^j$, $\beta_{i,k}^j$ and the standard deviation of the collision avoidance constraint $\Delta_{i,k}$ must satisfy the following:*

$$\alpha_{i,k}^j \triangleq -C^T(C\overline{\boldsymbol{x}}_i - \hat{\boldsymbol{y}}_{i,k}^j) \tag{5.10}$$

$$\beta_{i,k}^j \triangleq \overline{r}_k\|(C\overline{\boldsymbol{x}}_i - \hat{\boldsymbol{y}}_{i,k}^j)\|_2 - (C\overline{\boldsymbol{x}}_i)^T(C\overline{\boldsymbol{x}}_i - \hat{\boldsymbol{y}}_{i,k}^j) \tag{5.11}$$

$$\Delta_{i,k} \triangleq \sqrt{\boldsymbol{p}_i^T\Sigma_{\alpha_{i,k}}\boldsymbol{p}_i + 2\boldsymbol{p}_k^T\Sigma_{\alpha\beta_{i,k}} + \Sigma_{\beta i,k}}, \tag{5.12}$$

where

$$\Sigma_{\alpha_{i,k}} \triangleq cov\left(\alpha^j_{i,k}, \alpha^j_{i,k}\right) \tag{5.13}$$

$$\Sigma_{\beta_{i,k}} \triangleq cov\left(\beta^j_{i,k}, \beta^j_{i,k}\right) \tag{5.14}$$

$$\Sigma_{\alpha\beta_{i,k}} \triangleq cov\left(\alpha^j_{i,k}, \alpha^j_{i,k}\right) \tag{5.15}$$

are sample covariance matrices computed using the bootstrapped coefficients $\{\alpha_{i,k}\}_1^{N^{strap}}$ *and* $\{\beta_{i,k}\}_1^{N^{strap}}$ *and* $\boldsymbol{p}_i \triangleq \boldsymbol{C}\boldsymbol{x}_i \in \mathbb{R}^3$. *Let the dimension of the null space of* $\Sigma_{\alpha_{i,k}}$ *be* $n_{i,k} \geq 0$. [1] *The standard deviation* $\Delta_{i,k}$ *has the following upper bound,*

$$\Delta_{i,k} \leq \mathbf{1}^T|\tilde{\Sigma}^{\frac{1}{2}}_{\alpha_{i,k}}\left(\boldsymbol{p}_i - \boldsymbol{h}_{i,k}\right)| + \sqrt{3k_{i,k}} \triangleq \zeta_{i,k}, \tag{5.16}$$

where

$$\tilde{\Sigma}_{\alpha_{i,k}} = \Sigma_{\alpha_{i,k}} + I^{null}_{i,k} \tag{5.17}$$

$$I^{null}_{i,k} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{n_{i,k}\times n_{i,k}} \end{bmatrix} \in \mathbb{R}^{3\times 3} \tag{5.18}$$

and

$$\begin{bmatrix} \boldsymbol{h}_{i,k} \\ k_{i,k} \end{bmatrix} \triangleq \begin{bmatrix} -\left(\Sigma_{\alpha_{i,k}} + I^{null}_{i,k}\right)^{-1}\Sigma_{\alpha\beta_{i,k}} \\ \Sigma_{\beta_{i,k}} - \Sigma^T_{\alpha\beta_{i,k}}\left(\Sigma_{\alpha_{i,k}} + I^{null}_{i,k}\right)^{-1}\Sigma_{\alpha\beta_{i,k}} \end{bmatrix}. \tag{5.19}$$

Proof. Let the eigendecomposition of $\Sigma_{\alpha_{i,k}}$ be the following:

$$\Sigma_{\alpha_{i,k}} = \begin{bmatrix} U_r & U_n \end{bmatrix} \begin{bmatrix} \Lambda_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_r & U_n \end{bmatrix}^T,$$

where $U_r \in \mathbb{R}^{3\times(3-n_{i,k})}$ is comprised of the eigenvectors of $\Sigma_{\alpha_{i,k}}$ that are orthonormal. The columns of $U_n \in \mathbb{R}^{n_{i,k}}$ are the complementary orthonormal basis that spans the null space of $\Sigma_{\alpha_{i,k}}$. By substituting (5.19) one can verify the following inequality:

$$\Delta_{i,k} \leq \sqrt{(\boldsymbol{p}_i - \boldsymbol{h}_{i,k})^T\tilde{\Sigma}_{\alpha_{i,k}}(\boldsymbol{p}_i - \boldsymbol{h}_{i,k}) + k_{i,k}} \triangleq \tilde{\Delta}_{i,k} \tag{5.20}$$

[1] In all numerical simulations, $\Sigma_{\alpha_{i,k}}$ is strictly positive definite. However, when one or more measurable states are noiseless, $\Sigma_{\alpha_{i,k}}$ can be ill-conditioned. Instead of adding $I^{\text{null}}_{i,k}$, which can be numerically expansive to determine, it is recommended to apply Algorithm 4 only to states with measurement noise and to adapt Theorem 6 with deterministic forecasts for the noiseless states and the DRCC formulation for the noisy ones.

where $\tilde{\Sigma}_{\alpha_{i,k}}$ is a positive definite matrix because

$$\tilde{\Sigma}_{\alpha_{i,k}} = \begin{bmatrix} U_r & U_n \end{bmatrix} \left(\begin{bmatrix} \Lambda_r & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & I_{n_{i,k} \times n_{i,k}} \end{bmatrix} \right) \begin{bmatrix} U_r & U_n \end{bmatrix}^T .$$

The desired upper bound (5.20) holds by adding a positive constant,

$$\boldsymbol{\iota}_{i,k} \triangleq \frac{2}{\sqrt{3}} \mathbf{1}^T |\tilde{\Sigma}^{1/2}_{\alpha_{i,k}} \boldsymbol{p}_k - \boldsymbol{h}_k|,$$

to $\tilde{\Delta}^2_{i,k}$ which satisfied the following inequalities

$$\tilde{\Delta}^2_{i,k} \leq \tilde{\Delta}^2_{i,k} + \frac{2}{\sqrt{3}} \mathbf{1}^T |\tilde{\Sigma}^{1/2}_{\alpha_{i,k}} (\boldsymbol{p}_k - \boldsymbol{h}_k)| \leq \sqrt{\xi^T_{i,k} \xi_{i,k}}$$

where

$$\xi_{i,k} \triangleq |\tilde{\Sigma}^{1/2}_{\alpha_{i,k}} (\boldsymbol{p}_k - \boldsymbol{h}_{i,k})| + \mathbf{1} \sqrt{\frac{k_{i,k}}{3}} \in \mathbb{R}^3.$$

For the inequality to hold, the expression $\boldsymbol{\iota}_{i,k}$ must always be non-negative which is true by construction. Further, let $\zeta_{i,k} = \mathbf{1}^T \xi_{i,k} \in \mathbb{R}$, then

$$\zeta^2_{i,k} = (\xi^T_{i,k} \mathbf{1})(\mathbf{1}^T \xi_{i,k}) = \xi^T_{i,k} \xi_{i,k} + 2\epsilon_\xi .$$

If $\epsilon_\xi \geq 0$, the condition $\Delta_{i,k} \leq \zeta_{i,k}$ is true which completes the proof. By construction

$$\xi_{i,k} = [\xi^x_{i,k}, \xi^y_{i,k}, \xi^z_{i,k}] \in \mathbb{R}^3,$$

one can show mathematically

$$\epsilon_\xi = \xi^x_{i,k} \xi^y_{i,k} + \xi^x_{i,k} \xi^z_{i,k} + \xi^y_{i,k} \xi^z_{i,k} > 0$$

because $\xi^x_{i,k}, \xi^y_{i,k}, \xi^z_{i,k} \in \mathbb{R}_{>0}$. □

It is costly to incorporate each boostrap as a separate obstacle constraint, as the number of constraints grow linearly with N^{strap}. Instead, the ensemble mean and covariance of the distance from the obstacle are estimated. A DRCC accounts for *all bootstrap distributions* that can have this mean and covariance. This approach results in a significantly fewer obstacle constraints, whose cardinality remains fixed regardless of the number of bootstrap predictions N^{strap}.

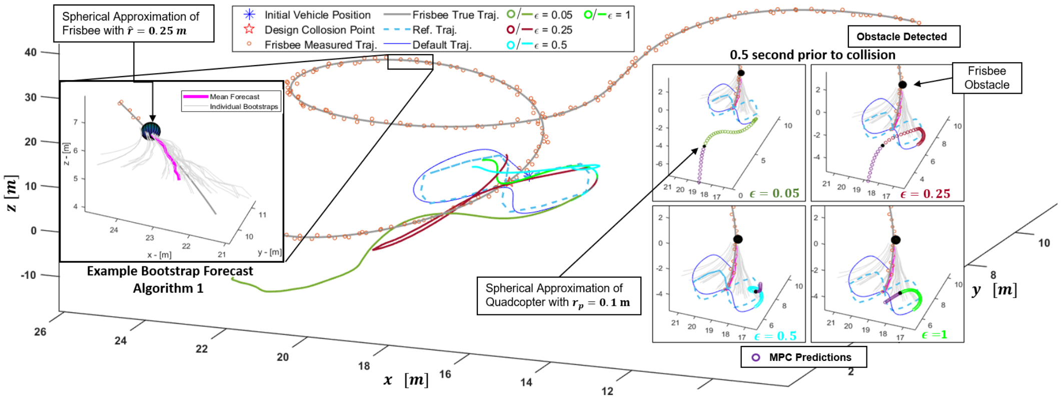

Figure 5.2: Four Monte-Carlo simulations with agent dynamics (5.24) and a Frisbee obstacle (see Fig.5.1) are compared. The same obstacle behaviors are simulated while the agent tracks the same figure ’8’ reference trajectory with four risk levels $\epsilon = \{0.05, 0.25, 0.5, 1\}$. The simulation is designed to be difficult: the vehicle must deviate from its reference trajectory as the obstacle trajectory is designed to intersect the agent’s reference trajectory with noise obstacle trajectory measurements. All measurement noises are sampled uniformly between $[-0.125, 0.125]$ meters. The bootstrap obstacle forecast uses the parameters: $L = 24$, $N^{\text{train}} = 100$, $N^{\text{step}} = 5$, $\delta_t = 20$, $N^{\sigma} = 8$, $N^{\text{strap}} = 40$. SSA-MPC uses the constants $n_h = 10$, $\chi = 50$ and $\tau = 0.25$ with fixed 4-step SCP iterations. The tuple $(\{\lambda^j\}_1^{t_j}, \{\boldsymbol{\mu}^j\}_1^{t_j}, \boldsymbol{\phi}^j), \forall j \in \mathbb{Z}^{1:40}$ in Algorithm 4 is computed with observables measured at 20 Hz. The four sub-diagrams show the planned trajectory at 4 risk levels; the planner is more conservative as $\epsilon \longrightarrow 0$, and aligns with the results shown in Table 5.1 and 5.2.

Theorem 6. *(SSA-MPC) Consider Problem 6 under Assumptions 4-6 with system dynamics* (5.1) *and bootstrap SSA forecasts of all obstacles’ center positions. Given a risk tolerance ϵ, the solution to the following optimal control problem is a feasible solution of Problem 6 as $w \longrightarrow \infty$. The SCP optimization problem at iteration w is:*

$$\{\boldsymbol{u}_i^*\}_1^{n_h} = \min_{\substack{\boldsymbol{u}_i \,\in\, \mathbb{R}^{n_{\boldsymbol{u}}} \\ \boldsymbol{s}_{i,k} \,\in\, \mathbb{R}^3}} \sum_{i=1}^{n_h} \mathcal{L}_i(\boldsymbol{y}_i^{ref} - \boldsymbol{G}\boldsymbol{x}_i, \boldsymbol{u}_i) \tag{5.21a}$$

$$s.t. \quad \boldsymbol{x}_{i+1} = \boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{B}\boldsymbol{u}_i \tag{5.21b}$$

$$\boldsymbol{x}_i \in \mathcal{D}^{\boldsymbol{x}}, \quad \boldsymbol{u}_i \in \mathcal{D}^{\boldsymbol{u}}, \quad \boldsymbol{x}_1 = \boldsymbol{x}_{init} \tag{5.21c}$$

$$\Lambda_{i,k} \begin{bmatrix} \boldsymbol{x}_i & \boldsymbol{s}_{i,k} \end{bmatrix}^T \leq \Gamma_{i,k} \tag{5.21d}$$

$$\|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i\| \leq \chi\tau^w \quad \forall (i,k) \in \mathbb{Z}_1^{n_h} \times \mathbb{Z}_1^{N^{obs}} \tag{5.21e}$$

where $\{\bar{\boldsymbol{x}}\}_1^{n_h}$ is the solution to the $(w-1)^{th}$ iteration of the SCP optimization,

$\Lambda_{i,k} \in \mathbb{R}^{7\times 11}$ *and* $\Gamma_{i,k} \in \mathbb{R}^7$ *encode the risk-based collision avoidance relationships,*

$$\Lambda_{i,k} = \begin{bmatrix} \mathbb{E}[\alpha_{i,k}]^T C & \mathbf{1}^T \nu_{\epsilon_n} \\ \tilde{\Sigma}^{1/2}_{\alpha_{i,k}} C & -I_{3\times 3} \\ -\tilde{\Sigma}^{1/2}_{\alpha_{i,k}} C & -I_{3\times 3} \end{bmatrix}, \quad \Gamma_{i,k} = \begin{bmatrix} -\mathbb{E}[\beta_{i,k}] - \nu_{\epsilon_n}\sqrt{3k_{i,k}} \\ \tilde{\Sigma}^{1/2}_{\alpha_{i,k}} \boldsymbol{h}_{i,k} \\ -\tilde{\Sigma}^{1/2}_{\alpha_{i,k}} \boldsymbol{h}_{i,k} \end{bmatrix},$$

where $\epsilon_n \triangleq \frac{\epsilon}{N^{obs}}$ *and* $\nu_{\epsilon_n} \triangleq \sqrt{\frac{1-\epsilon_n}{\epsilon_n}}$ *. Lastly,* $\chi \geq 0$ *is the initial trust region and* $\tau \in (0,1)$ *the worst-case rate of convergence.*

Proof. Denote the j^{th} random bootstrapped obstacle forecastsas

$$z^j_{i,k} \triangleq (\alpha^j_{i,k})^T \boldsymbol{x}_i + \beta^j_{i,k},$$

where $\alpha^j_{i,k}$ and $\beta^j_{i,k}$ are defined in (5.10) and (5.11). The obstacle avoidance constraint (5.8) has an affine over approximation (5.9), which is equivalently given by $z^j_{i,k} < 0$. Hence, the chance constraint (5.3d) is,

$$\mathbb{P}(\boldsymbol{x}_i \in \mathcal{O}_{\text{pred}}) = \mathbb{P}\Big(\bigcup_{k=1}^{N^{\text{obs}}} \{z_{i,k} \geq 0\}\Big) \leq \sum_{k=1}^{N^{\text{obs}}} \mathbb{P}(z_{i,k} \geq 0).$$

Enforcing $\mathbb{P}(z_{i,k} \geq 0) \leq \epsilon_n, \forall k \in \mathbb{Z}^{1:N^{\text{obs}}}$ also satisfies (5.3d). This constraint satisfaction is equivalent to a DRCC:

$$\sup_{\boldsymbol{\kappa} \sim \left(\mathbb{E}[z_{i,k}], \Sigma_{z_{i,k}}\right)} \mathbb{P}\{\boldsymbol{\kappa} \geq 0\} \leq \epsilon_n, \forall i, k \in \mathbb{Z}^{n_h}_1 \times \mathbb{Z}^{N^{\text{obs}}}_1,$$

where $\mathbb{E}[z_{i,k}]$ and $\Sigma_{z_{i,k}}$ are the sample mean and covariance of the bootstrapped $\{z_{i,j,k}\}_{j=1}^{N^{\text{strap}}}$. The above statement can be reformulated as a deterministic constraint as shown in [171],

$$\underbrace{\mathbb{E}[z_{i,k}]}_{\mathbb{E}[\alpha_{i,k}]^T C\boldsymbol{x}_i + \mathbb{E}[\beta_{i,k}]} + \nu_{\epsilon_n} \underbrace{\sqrt{\Sigma_{z_{i,k}}}}_{\Delta_{i,k}} \leq 0, \forall i \in \mathbb{Z}^{1:n_h}, k \in \mathbb{Z}^{N^{\text{obs}}}_1. \tag{5.22}$$

Constraint (5.22) is not affine in the optimization variable, as is desirable for real-time application. By Lemma 4, $\Delta_{i,k} \leq \zeta_{i,k}$, and the following tighter inequality constraint can be deduced as a numerically appealing alternative to (5.22),

$$\mathbb{E}[\alpha_{i,k}]^T C\boldsymbol{x}_i + \mathbb{E}[\beta_{i,k}] + \nu_{\epsilon_n}\left(\mathbf{1}^T|\Sigma^{1/2}_{\alpha_{i,k}}\boldsymbol{p}_i - \boldsymbol{h}_{i,k}| + \sqrt{3k_{i,k}}\right) \leq 0. \tag{5.23}$$

To account for the absolute value term, the auxiliary optimization variables $\boldsymbol{s}_{i,k}$ are introduced that satisfy the following inequalities:

$$\Sigma_{\alpha_{i,k}}^{1/2}\boldsymbol{p}_i - \boldsymbol{h}_{i,k} \leq \boldsymbol{s}_{i,k}, \quad -\Sigma_{\alpha_{i,k}}^{1/2}\boldsymbol{p}_i + \boldsymbol{h}_{i,k} \leq \boldsymbol{s}_{i,k}.$$

Therefore, satisfying (5.21d) is equivalent to satisfying (5.23).

Convergence of the SCP is proven in [172] which is based on implementing a trust region via second-order cone constraints (5.21e). The authors also show the solution to the SCP as $w \to \infty$ is a feasible solution to Problem 6. To be numerically feasible, w is usually bounded above by a finite integer, resulting in a sub-optimal but still feasible solution.

□

5.5 Results and Discussion

Considering the problem of a quadcopter follows a reference trajectory $\boldsymbol{y}^{\mathrm{ref}}$ while avoiding unknown moving obstacles and respecting state and control constraints. Its position and Euler angles (roll, pitch, yaw) in frame E are denoted $x, y, z, \varphi, \theta, \psi$.

Example 1: Fully-actuated multirotor with attitude controller

Assume there exists a low-level attitude controller that tracks given attitude commands within 20 Hz. As a result, the following linear dynamic model is employed to extract a high-level motion planner that outputs attitude and thrust inputs:

$$\ddot{x} = -g\theta, \;\; \ddot{y} = g\varphi, \;\; \ddot{z} = u_1 - g, \;\; \dot{\psi} = u_4, \tag{5.24}$$

where the planner control inputs are given by $u_1, \theta, \varphi, u_4$ which are thrust, roll angle, pitch angle, and yaw rate, and where $g = 9.81 m/s^2$ is the gravitational acceleration.

Example 2: Multirotor operating in small angle regime

A mixer maps thrust and moment inputs into electronic speed controller PWM commands at 8kHz. Since the multirotor is constrained to operate within the state constraints $\theta \in [-0.45, 0.45]$ radians and $\varphi \in [-0.45, 0.45]$ radians, the following standard multirotor linear dynamics is used,

$$\ddot{x} = -g\theta, \;\; \ddot{y} = g\varphi, \;\; \ddot{z} = u_1 - g, \tag{5.25}$$

$$\ddot{\varphi} = \frac{u_2}{I_{xx}}, \; \ddot{\theta} = \frac{u_3}{I_{yy}}, \ddot{\psi} = \frac{u_4}{I_{zz}}, \tag{5.26}$$

where the planner control inputs u_1, u_2, u_3, u_4 corresponds to the thrust force in the body frame and three moments. The vehicle's moments of inertia are $I_{xx} = 0.0075\ kgm^2, I_{yy} = 0.0075\ kgm^2, I_{zz} = 0.013\ kgm^2$.

For both examples, the desired reference trajectory consists of positions, $\{x^{ref}\}$, $\{y^{ref}\}$, $\{z^{ref}\}$ and yaw angles $\{\psi^{ref}\}$.

Monte-Carlo (MC) simulations of the proposed planner are conducted as it avoids three differently behaved obstacles which are introduced once in each run for both examples. Case 1 uses a constant-speed spherical obstacle without drag. Case 2 is a thrown spherical obstacle with drag. In Case 3, a Frisbee is thrown at various initial positions, velocities, and rotation speed. The sphere dynamics are captured by a 6-state ODE with drag penalties proportional to its velocities. The Frisbee is modeled following [167], using a 12-state model identical to Fig. 5.1 with aerodynamic drag. The Frisbee is modeled as a sphere with the same radius as the Frisbee disk. The method was compared against an artificial potential field alternative. A supplementary video (`https://youtu.be/6s8pfRZ171Q`) provides more details.

A total of 1000 MC simulations per ϵ level are conducted to compare the numerical feasibility, percent success in obstacle avoidance (if the MPC planner is feasible), and the planner's conservativeness, as measured by the minimum distance between the obstacle and agent centers. For the three cases, the obstacle speed ranges are [0.41, 8.43], [3.41, 6.37], and [5.76, 6.68] m/s, respectively. The MPC planning and measurement rates are fixed at 20 Hz. With a 10-step horizon and 40 bootstraps, the average per planner update rate is 0.030 ± 0.0014 sec, using [173] on an Intel i7-9700K CPU @3.6GHz processor, using a dynamic simulation written in MATLAB.

The results in Tables 5.1 and 5.2 show the applicability of the SSA-MPC algorithm, despite vast differences in obstacle behavior. Further, as the risk tolerance ϵ shrinks, the percentage success in obstacle avoidance (when the solution is feasible) increases, with a trade-off in the feasibility of optimization (5.21). Parameters d_{min} and $\sigma(d_{min})$ are the average minimum distance between the agent and the obstacle and the standard deviation of this minimum distance across 1000 MC simulations, respectively. Based on Table 5.1 and 5.2, the risk tolerance ϵ can also viewed as a robustness parameter which is inversely proportional to the distance between the agent and obstacles. However, the cost of more robustly avoiding the obstacles is reflected in the numerical feasibility, a parameter describing the chances of the SCP formulation (5.21) being feasible for the entire simulation. The feasible set of the polytopic collision avoidance constraints (5.21d) shrinks as ϵ (and hence ϵ_n) decreases.

There are limitations of this approach. It degrades as more noise is injected into the system, which calls for more robust denoising methods than hard thresholding. The horizon length also matters. Note, that insufficient denoising may lead to a poor bootstrap forecast that compounds with increasing horizon length. A similar phenomenon is also observed in [174]. While risk-aware MPC handles a poor forecast internally, it could lead to numerical infeasibility.

Summary and Future Work

The proposed data-driven, risk-aware obstacle avoidance planner showcased near-perfect results in avoiding moving obstacles with limited, noisy measurements and no prior knowledge about the obstacle behaviors. This work offers a new paradigm that can extract obstacle dynamics online in a reasonable time and a compatible risk-

Cases	ϵ	0.05	0.1	0.25	0.5	0.75	1
	%Feas.	97.5	98.2	98.9	99.6	99.9	100
Const.	%Succ.	100	100	100	100	100	59.0
Speed	$\overline{d}_{min}$	2.26	1.85	1.41	1.12	0.94	0.64
	$\sigma(d_{min})$	0.42	0.33	0.25	0.22	0.24	0.35
	%Feas.	99.5	99.6	99.9	100	100	100
Ball	%Succ.	100	100	100	100	100	79.3
w/drag	$\overline{d}_{min}$	2.60	2.14	1.63	1.27	1.04	0.64
	$\sigma(d_{min})$	1.08	0.93	0.70	0.50	0.37	0.27
	%Feas.	90.3	97.4	98.3	98.6	97.5	97.8
Frisbee	%Succ.	100	100	100	100	99.7	58.0
w/drag	$\overline{d}_{min}$	4.97	3.97	2.85	2.01	1.44	0.78
	$\sigma(d_{min})$	1.97	1.53	1.15	0.91	0.76	0.77

Table 5.1: Summary of results from MC simulations of system (5.24)

Cases	ϵ	0.05	0.1	0.25	0.5	0.75	1
	%Feas.	83.9	84.2	85.6	86.5	88.6	99.8
Const.	%Succ.	100	100	100	100	100	61.2
Speed	$\overline{d}_{min}$	10.42	8.67	6.59	4.88	3.57	0.46
	$\sigma(d_{min})$	2.47	2.28	1.86	1.43	1.16	0.14
	%Feas.	92.8	91.8	90.9	91.1	90.7	82.8
Ball	%Succ.	100	100	100	100	100	0.1
w/drag	$\overline{d}_{min}$	6.95	5.82	4.25	3.04	2.13	1.82
	$\sigma(d_{min})$	2.52	2.55	2.27	1.98	1.59	N/A
	%Feas.	92.2	94.5	92.9	92.3	86.9	100
Frisbee	%Succ.	100	100	100	100	100	40.5
w/drag	$\overline{d}_{min}$	12.5	11.3	9.10	6.59	4.24	0.34
	$\sigma(d_{min})$	2.47	2.84	3.00	2.81	2.27	0.05

Table 5.2: Summary of results from MC simulations of system (6.9)

aware MPC formulation that also works in real-time. However, there are limitations in the current formulation that suggest future work. As a major limitation, the use of a linear MPC formulation significantly restricts the multirotor from performing agile maneuvers. Appealing to nonlinear model predictive methods could reduce the numerical infeasibility due to roll and pitch state constraints. In terms of computational efficiency, the number of auxiliary optimization variables $\boldsymbol{s}_{i,k}$ and the number of collision-avoiding inequality constraints all scale linearly with the number of obstacles. There is significant room for improvement of this efficiency by assigning obstacle priorities. This approach also needs to be benchmarked in high-fidelity environments like GAZEBO and/or validated by hardware experiments in the future. The simulation results show that adjusting the risk level, which is also the probability of constraint violation, ϵ can implicitly adjust the safety distance between the agent and obstacles. This observation can be further strengthened by empirical correlations and/or theoretical results.

Chapter 6

MOA: UNCERTAINTY-INFORMED CONTROL

Motion planning of autonomous systems in dynamic environments requires the system to reason about uncertainty in its environment, *e.g.*, a self-driving car needs to reason about uncertainty in the motion of other vehicles, and a mobile robot navigating a crowded space needs to assess the uncertainty of nearby pedestrians. These applications are safety-critical, as the agents' intentions are unknown, and systems must be able to plan reactive behaviors in response to an increase in uncertainty.

Chapter 5 shows a method that solves the challenging problem of extracting a dynamics model from real-time attainable data that is limited, noisy, and partial. Yet, it is difficult to prove prediction accuracy from these data-driven models since they are effectively out-of-sample extrapolations without ground truth for model calibration. Therefore, it is essential from a safety perspective to understand the uncertainty associated with the predicted obstacle trajectories.

This chapter details an algorithm that adaptively quantifies the uncertainty of trajectory predictions using adaptive conformal prediction. This algorithm is distribution-free and applies to a broad class of trajectory predictors, providing probabilistic coverage on average. Leveraging the quantified prediction uncertainties, a pairing model predictive controller is proposed to plan probabilistically safe paths around dynamic obstacles.

This chapter was adapted from: [14]

6.1 Problem Statement

The discrete-time dynamical system governs the dynamics of the autonomous system,

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t), \tag{6.1}$$

where $\boldsymbol{x}_t \in \mathcal{D}^{\boldsymbol{x}} \subseteq \mathbb{R}^{n_x}$ and $\boldsymbol{u}_t \in \mathcal{D}^{\boldsymbol{u}} \subseteq \mathbb{R}^{n_u}$ denote the state and the control input at time $t \in \mathbb{Z}_{\geq 0}$, respectively. The sets $\mathcal{D}^{\boldsymbol{u}}$ and $\mathcal{D}^{\boldsymbol{x}}$ denote the set of permissible control inputs and the workspace of the system, respectively. The measurable function $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ describes the system dynamics. Let $\boldsymbol{x}_0 \in \mathcal{D}^{\boldsymbol{x}}$ be the initial condition of the system. For brevity, let $X \triangleq [\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots]$ denote the trajectory of (6.1) under a given control sequence $U \triangleq [\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots]$.

The system operates in an environment with N dynamic agents whose trajectories are *a priori* unknown. Let $\mathcal{D}$ be an unknown distribution over agent trajectories, *i.e.*, let $Y \triangleq [Y_0, Y_1, \ldots] \sim \mathcal{D}$ describe a random trajectory where the joint agent state $Y_t \triangleq (Y_{t,1}, \ldots, Y_{t,N})$ at times $t \in \mathbb{N} \cup \{0\}$ is drawn from $\mathbb{R}^{Nn_x}$, *i.e.*, $Y_{t,j}$ is the state of agent j at time t. For instance, Y_t can denote the uncertain two-dimensional positions of N pedestrians at time t. Modeling dynamic agents by a distribution Δ provides great flexibility, and Δ can generally describe the motion of Markov decision processes.

There are no other assumptions on the distribution Δ, and in the proposed algorithm predicts states $(Y_{t+1}, \ldots, Y_{t+n_h})$ for a prediction horizon of n_h from $(Y_0, \ldots, Y_t)$ and quantifies prediction uncertainty using ideas from ACP.

Problem 4. *Given the system in* (6.1)*, the unknown random trajectories* $Y \sim \Delta$*, and a failure probability* $\epsilon \in (0, 1)$ *(referred to as risk level or risk tolerance), design the control inputs* $\boldsymbol{u}_t$ *such that the Lipschitz continuous constraint function* $c : \mathbb{R}^{n_x} \times \mathbb{R}^{Nn_x} \to \mathbb{R}$ *is satisfied*[1] *with a probability of at least* $1 - \epsilon$ *on average, i.e., that*

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{P}\left[c(\boldsymbol{x}_t, \boldsymbol{y}_t) \geq 0\right] \geq 1 - \epsilon. \tag{6.2}$$

Note, the prior work ([175]) considers a similar problem formulation, but where in this case of pointwise satisfaction of the constraint function c is desired, *i.e.*, $\mathbb{P}\left[c(\boldsymbol{x}_t, Y_t) \geq 0\right] \geq 1 - \epsilon$ for all $t \geq 0$. In [175], however, it is assumed that an offline calibration data set drawn from Δ is available, which is not being assumed here. This makes the problem more challenging so it is desirable for the average probabilistic guarantee in equation (6.2). Additionally, this formulation allows the distribution Δ to depend on $\boldsymbol{x}$, *e.g.*, a pedestrian's behavior may change if a car comes too close, which is not possible in [175]. This work is a step towards the implementation of a general framework that can adapt to such changes in the agent distribution.

To address Problem 4, trajectory predictors are used to predict the motion of the agents $(Y_0, Y_1, \ldots)$ to enforce the constraint (6.2) within a MPC framework. In

[1] For an obstacle avoidance constraint, like $c(\boldsymbol{x}, \boldsymbol{y}) \triangleq \|\boldsymbol{x} - \boldsymbol{y}\| - 0.5 \geq 0$, the Lipschitz constant is 1. It is implicitly assumed that the constraint function is initially satisfied, *i.e.*, that $c(\boldsymbol{x}_0, \boldsymbol{y}_0) \geq 0$.

[175], The availability of calibration data from $\mathcal{D}$ is assumed to build prediction regions that quantify the uncertainty of trajectory predictors. In this setting, one can collect data online to adapt the uncertainty sets based on the past performance of the predictor using ACP without any assumptions on the distribution of the uncertainty and exchangeability of the validation and training data set.

Trajectory Predictors: Given observations $(Y_0, \ldots, Y_t)$ at time t, the objective of the trajectory predictor is to predict future states $(Y_{t+1}, \ldots, Y_{t+n_h})$ for a prediction horizon of n_h. Assume that Predict is a function that maps observations $(Y_0, \ldots, Y_t)$ to predictions $(\hat{Y}_t^1, \ldots, \hat{Y}_t^{n_h})$ of $(Y_{t+1}, \ldots, Y_{t+n_h})$. Note that t in $\hat{Y}_t^\tau$ denotes the time at which the prediction is made, while τ indicates how many future steps are predicted. In principle, Predict can be a classical auto-regressive model or a neural network based method.

While the proposed problem solution is compatible with any trajectory predictor Predict, this work studies the case where real-time updating strategies like sliding linear predictors with extended Kalman filter. Extracting a dynamics model from data is challenging, especially when the available data is limited, noisy, and partial. [66] showed that the method of delays can be used to reconstruct qualitative features of the full-state, phase space from delayed partial observations. By building on the prior work (Chapter 5) using time delay embedding in dynamic obstacle avoidance, a linear predictor based on spatial-temporal factorization of the delayed partial observations can be used as the pairing trajectory predictor.

6.2 Adaptive Conformal Prediction Regions

Recall that the predictions $(\hat{Y}_t^1, \ldots, \hat{Y}_t^{n_h})$ are not measurable at time t of future agent states $(Y_{t+1}, \ldots, Y_{t+n_h})$ from past observations $(Y_0, \ldots, Y_t)$ using the Predict function. Note, however, that these point predictions contain no information about prediction uncertainty and can hence not be used to reason about the safety constraint (6.2). To tackle this issue, prediction regions for $(Y_{t+1}, \ldots, Y_{t+n_h})$ is constructed using ideas from ACP, reviewed in Section 2.3.

The nonconformity score $\|Y_{t+\tau} - \hat{Y}_t^\tau\|$ at time t is consider to obtain prediction regions for $(Y_{t+1}, \ldots, Y_{t+n_h})$ that captures the multi step-ahead prediction error for each $\tau \in \{1, \ldots, n_h\}$. A large nonconformity score indicates that the prediction $\hat{Y}_t^\tau$ of $Y_{t+\tau}$ is not accurate, while a small score indicates an accurate prediction. For each τ, it is desirable to obtain a prediction region using C_t^τ that is again defined by an update variable ϵ_t^τ. Note, however, that the quantity $\|Y_{t+\tau} - \hat{Y}_t^\tau\|$ can not be evaluated at time t as only measurements $(Y_0, \dots, Y_t)$ are known, but not $(Y_{t+1}, \dots, Y_{t+n_h})$. Consequently, the update rule (2.12) is not applicable to update ϵ_t^τ, as the prediction error would depend on checking if $\|Y_{t+\tau} - \hat{Y}_t^\tau\| \leq C_t^\tau$. To address this issue, we define the time-lagged nonconformity score as

$$R_t^\tau \triangleq \|Y_t - \hat{Y}_{t-\tau}^\tau\|$$

that this quantity is available at time t using the update rule (2.12). This nonconformity score R_t^τ is time-lagged in the sense that, at time t, the τ step-ahead prediction errors are assessed that were made τ time steps ago. Now, by update the parameter ϵ_{t+1}^τ, the τ-lagged prediction region is defined C_{t+1}^τ as

$$\epsilon_{t+1}^\tau \triangleq \epsilon_t^\tau + \gamma(\epsilon - \mathbf{1}_{\|Y_t - \hat{Y}_{t-\tau}^\tau\| \geq C_t^\tau}). \tag{6.3}$$

Let C_{t+1}^τ be the $\lceil (t - \tau + 1)(1 - \epsilon_{t+1}^\tau) \rceil^{\text{th}}$ smallest value of $(R_\tau^\tau, \dots, R_t^\tau)$[2].

By obtaining a prediction region for R_{t+1}^τ using ACP, a prediction region for the τ step-ahead prediction error can be computed that was made $\tau - 1$ time steps ago, *i.e.*, for $\|Y_{t+1} - \hat{Y}_{t+1-\tau}^\tau\|$. Under the assumption that R_{t+1}^τ and $R_{t+\tau}^\tau$ are independent and identically distributed, R_{t+1}^τ serves as a prediction region for τ step-ahead prediction error that was made 0 time steps ago (now at time t), *i.e.*, for $R_{t+\tau}^\tau$ which encodes $\|Y_{t+\tau} - \hat{Y}_t^\tau\|$. Naturally, in the setting R_{t+1}^τ and $R_{t+\tau}^\tau$ are not independent and identically distributed. However, by adapting the prediction regions based on the performance of the multi-step predictions using ACP, on-average coverage guarantees are obtained, albeit weaker than the point-wise guarantees provided by conformal predictions. It is important to note that for the theoretical guarantees provided in the next section, only the one-step-ahead prediction errors are relevant.

Corollary 2. *Let γ be a learning rate, $\epsilon_0^1 \in (0, 1)$ be an initial value for the recursion* (7.42)*, and T be the number of times that the recursion* (7.42) *is computed. Then, for the one-step-ahead prediction errors, it holds that*

$$1 - \epsilon - p_1 \leq \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{P}\left[\|Y_{t+1} - \hat{Y}_t^1\| \leq C_{t+1}^1\right] \leq 1 - \epsilon + p_2 \tag{6.4}$$

with constants

$$p_1 \triangleq \frac{\epsilon_0^1 + \gamma}{T\gamma},$$

[2] Instead of keeping track of all data, a sliding window of the N most recent data is selected to generate the nonconformity scores. For all prediction regions, it is recommended to consider $(R_{t-N}^\tau, \dots, R_t^\tau)$ and to compute C_{t+1}^τ as the $\lceil (N + 1)(1 - \epsilon_{t+1}^\tau) \rceil^{\text{th}}$ smallest value.

$$p_2 \triangleq \frac{(1-\epsilon_0^1)+\gamma}{T\gamma}$$

so that $\lim_{T\to\infty} p_1 = 0$ *and* $\lim_{T\to\infty} p_2 = 0$.

Proof. Since the probability of an event is equivalent to the expected value of the indicator function of that event, it follows by the definition of the error $\mathbf{1}_{\|Y_{t+1}-\hat{Y}_t^1\|\geq C_{t+1}^1}$ that

$$\mathbb{P}\left[\|Y_{t+1}-\hat{Y}_t^1\| \leq C_{t+1}^1\right] = \mathbb{E}[1-\mathbf{1}_{\|Y_{t+1}-\hat{Y}_t^1\|\geq C_{t+1}^1}] = 1-\mathbb{E}[\mathbf{1}_{\|Y_{t+1}-\hat{Y}_t^1\|\geq C_{t+1}^1}]. \quad (6.5)$$

For a given initialization ϵ_0^τ and learning rate γ, [109, Proposition 4.1] showed that the following bound holds (with probability one) for the misclassification errors

$$\frac{-(1-\epsilon_0^1)+\gamma}{T\gamma} \leq \frac{1}{T}\sum_{t=0}^{T-1}\mathbf{1}_{\|Y_{t+1}-\hat{Y}_t^1\|\geq C_{t+1}^1} - \epsilon \leq \frac{\epsilon_0^1+\gamma}{T\gamma}$$

implies

$$\left|\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{1}_{\|Y_{t+1}-\hat{Y}_t^1\|\geq C_{t+1}^1} - \epsilon\right| \leq \frac{\max(\epsilon_0^1, 1-\epsilon_0^1)+\gamma}{T\gamma}.$$

Hence, taking the expectation of the above two-sided inequality, the following inequalities hold

$$\begin{aligned}
&\frac{-(1-\epsilon_0^1)+\gamma}{T\gamma} \leq \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\mathbf{1}_{\|Y_{t+1}-\hat{Y}_t^1\|\geq C_{t+1}^1}] - \epsilon \leq \frac{\epsilon_0^1+\gamma}{T\gamma},\\
\overset{(a)}{\Leftrightarrow}\quad &\frac{-(1-\epsilon_0^1)+\gamma}{T\gamma} \leq \frac{1}{T}\sum_{t=0}^{T-1}\left(1-\mathbb{P}\left[\|Y_{t+1}-\hat{Y}_t^1\| \leq C_{t+1}^1\right)\right] - \epsilon \leq \frac{\epsilon_0^1+\gamma}{T\gamma},\\
\Leftrightarrow\quad 1-\epsilon+&\frac{(1-\epsilon_0^1)+\gamma}{T\gamma} \geq \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{P}\left[\|Y_{t+1}-\hat{Y}_t^1\| \leq C_{t+1}^1\right] \geq 1-\epsilon-\frac{\epsilon_0^1+\gamma}{T\gamma},
\end{aligned}$$

where equation (6.5) is used for getting the equivalence in (a). □

Remark 4. *The above result can be similarly extended to the FACP case with a set of candidate learning rates, γ, in [111, Theorem 3.2].*

To illustrate these multistep-ahead prediction regions, consider a planar double pendulum whose dynamics are governed by chaotic, nonlinear dynamics that are sensitive to the initial condition [176]. The predictions made by a linear predictor that uses noisy observations of the position of the double pendulum [177, Appendix] are studied where ACP is used to predict the uncertainty in the predictions. Both the trajectory predictor and the uncertainty quantification using ACP use online data from a single trajectory. ACP provides the multi-step errors in the linear predictions with a coverage level of $\epsilon = 0.1$, and learning rates $\gamma = \begin{pmatrix} 0.0008 & 0.0015 & 0.003 & 0.005 & 0.009 & 0.017 & 0.03 & 0.05 & 0.08 \end{pmatrix}$.

Figure 6.1 compares the 1-step and 6-step ahead error prediction regions to the true multi-step errors for two states, the second mass position, x_2, y_2. The percentages of one-step errors that are incorrectly predicted, *i.e.*, $\mathbf{1}_{\|Y_t - \hat{Y}^1_{t-1}\| \geq C^1_t} = 1$, for the positions of each mass, x_1, x_2, y_1, y_2 are $2.36\%, 0.94\%, 1.57\%, 1.73\%$, respectively. The effects of adaptation as the ACP prediction regions are observed to be larger in areas of poor performance of the linear predictor (and consequently higher error in the prediction) and smaller in regions where the linear predictor performs well. Note, the miscoverage levels are lower than the expected miscoverage when $\epsilon = 0.1$. This is because the learning rates γ used are small, because of which the adaptation is slower. As the learning rate becomes larger, the adaptation is faster and the miscoverage levels will be closer to 10%. In practice, higher learning rates lead to high variations in the prediction sets.

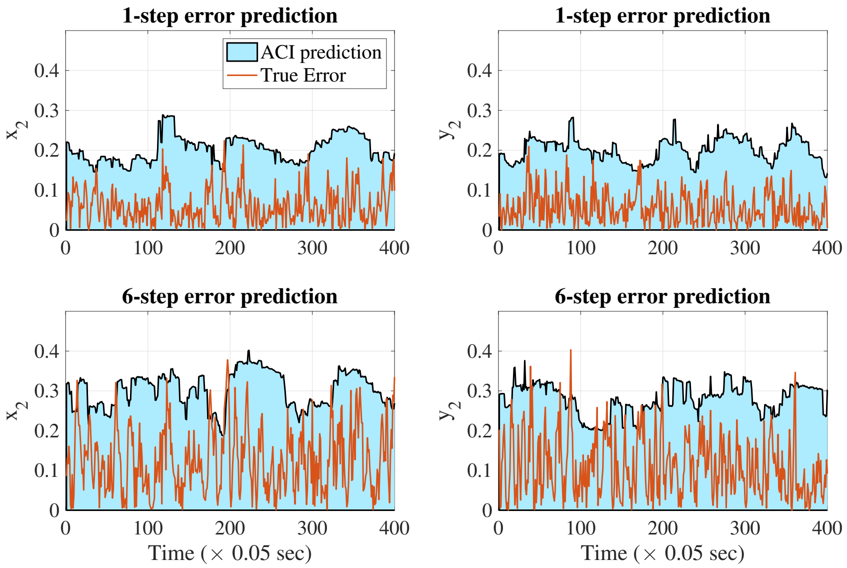

Figure 6.1: The multi-step prediction errors are shown for two of the six states of a double pendulum (x_2, y_2). ACP can correctly predict regions of high and low error (90% coverage regions) by adjusting the prediction quantile using update law (2.12). The orange lines are the true multi-step prediction errors and the blue areas are the error regions predicted by ACP.

6.3 Uncertainty-Informed MPC

Based on the obtained uncertainty quantification from the previous section an uncertainty-informed model predictive controller (UI-MPC) is proposed that uses predictions $\hat{Y}_t^\tau$ and adaptive predictions C_{t+1}^τ. The underlying optimization problem that is solved at every time step t is:

$$\min_{(\boldsymbol{u}_t,\ldots,\boldsymbol{u}_{t+n_h-1})} \sum_{k=t}^{t+n_h-1} \mathcal{L}(\boldsymbol{x}_{k+1}, \boldsymbol{u}_k) \tag{6.6a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k), \qquad k \in \{t, \ldots, t+n_h-1\} \tag{6.6b}$$

$$c(\boldsymbol{x}_{t+\tau}, \hat{Y}_t^\tau) \geq LC_{t+1}^\tau, \qquad \tau \in \{1, \ldots, n_h\} \tag{6.6c}$$

$$\boldsymbol{u}_k \in \mathcal{D}^{\boldsymbol{u}}, \boldsymbol{x}_{k+1} \in \mathcal{D}^{\boldsymbol{x}}, \qquad k \in \{t, \ldots, t+n_h-1\} \tag{6.6d}$$

where L is the Lipschitz constant of the constraint function c, $\mathcal{L}$ is a step-wise cost function, and $\{\boldsymbol{u}_t, \ldots, \boldsymbol{u}_{t+n_h-1}\}$ is the control sequence. The optimization problem in (6.6) is convex if the functions $\mathcal{L}$ and f are convex, the function c is convex in its first argument, and the sets $\mathcal{D}^{\boldsymbol{u}}$ and $\mathcal{D}^{\boldsymbol{x}}$ are convex.

Based on this optimization problem, a receding horizon control strategy is developed in Algorithm 5. Line 1 of Algorithm 5 initializes the parameter ϵ_0^t simply to ϵ. Lines 2-11 present the real-time planning loop:

1. Updating the states $\boldsymbol{x}_t$ and calculating new predictions $\hat{Y}_t^\tau$ (lines 3-4).
2. computing the adaptive nonconformity scores C_{t+1}^τ (lines 5-9), and 3) solving the optimization problem in (6.6) of which only $\boldsymbol{u}_t$ is applied (lines 10-11).

Remark 5. *While Algorithm 5 uses a single learning rate, one can similarly extend the above algorithm to be fully adaptive using a candidate set of* $\{\gamma_i\}_{1 \leq i \leq k}$ *without loss of generality.*

Remark 6. *[109] assume that when* $\epsilon_{t+1} \leq 0$*,* $C_{t+1} \to \infty$*. This means that when the algorithm requires robust behavior, the* ∞*-prediction region ensures that any prediction at the next time-step should be correctly classified. For a physical system, there are limits on how much the dynamic obstacle can accelerate in one time-step which gives an upper bound* $R_{max} < \infty$ *on the worst-case error. In practice, in equality* $0 \leq \epsilon_{t+1} \leq 1$ *should be enforced with* $C_{t+1} \leq R_{max}$*.*

Theorem 7. *Let* γ *be a learning rate,* $\epsilon_0^1 \in (0, 1)$ *be an initial value for the recursion* (7.42)*, and* T *be the number of times that the recursion* (7.42) *is computed. If the*

Algorithm 5: Uncertainty Informed MPC (UI-MPC)

```
Data: Failure probability $\epsilon$, prediction horizon $n_h$, learning rate $\gamma$
Result: Control input $\boldsymbol{u}_t(\boldsymbol{x}_t, Y_0, \dots, Y_t)$ at each time $t$
$\epsilon_0^\tau \leftarrow \epsilon$ for $\tau \in \{1, \dots, n_h\}$.
for $t$ from $0$ to $\infty$ do
    #real-time motion planning loop
    Update $\boldsymbol{x}_t$ and $Y_t$
    Obtain predictions $\hat{Y}_t^\tau$ for $\tau \in \{1, \dots, n_h\}$
    for $\tau$ from $1$ to $n_h$ do
        # compute ACP regions
        $\epsilon_{t+1}^\tau \leftarrow \epsilon_t^\tau + \gamma(\epsilon - \mathbf{1}_{\|Y_t - \hat{Y}_{t-\tau}^\tau\| \geq C_t^\tau})$
        Get $R_t^\tau \|Y_t - \hat{Y}_{t-\tau}^\tau\|$
        $q \leftarrow \lceil (t+1)(1 - \epsilon_{t+1}^\tau) \rceil$
        Set $C_{t+1}^\tau$ as the $q$th smallest value of $(R_\tau^\tau, \dots, R_t^\tau)$
    end
    Calculate controls $\{\boldsymbol{u}_t, ..., \boldsymbol{u}_{t+n_h-1}\}$ as the solution of (6.6)
    Apply $\boldsymbol{u}_t$ to (6.1)
end
```

optimization problem (6.6) *in Algorithm 5 is recursively feasible, then Algorithm 5 will lead to*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{P}\big(c(\boldsymbol{x}_{t+1}, Y_{t+1}) \geq 0\big) \geq 1 - \epsilon - p_1 \tag{6.7}$$

with constant $p_1 \triangleq \frac{\epsilon_0^1 + \gamma}{T\gamma}$ *so that* $\lim_{T\to\infty} p_1 = 0$.

Proof. By assumption, the optimization problem in (6.6) is feasible at each time $t \in \{0, 1, \dots\}$. Due to constraint (6.6c) and Lipschitz continuity of c, it hence holds that

$$0 \leq c(\boldsymbol{x}_{t+1}, \hat{Y}_t^1) - LC_{t+1}^1 \leq c(\boldsymbol{x}_{t+1}, Y_{t+1}) + L\|Y_{t+1} - \hat{Y}_t^1\| - LC_{t+1}^1 \tag{6.8}$$

at each time $t \in \{0, 1, \dots\}$. Consequently, note that $\|Y_{t+1} - \hat{Y}_t^1\| \leq C_{t+1}^1$ is a sufficient condition for $c(\boldsymbol{x}_{t+1}, Y_{t+1}) \geq 0$. In a next step, the followings can be derived

$$\begin{aligned}
&\mathbb{P}\left[c(\boldsymbol{x}_{t+1}, Y_{t+1}) \geq 0\right] \\
&\overset{(a)}{=} \mathbb{P}\left[c(\boldsymbol{x}_{t+1}, Y_{t+1}) \geq 0 \,\middle|\, \|Y_{t+1} - \hat{Y}_t^1\| \leq C_t^1\right] \mathbb{P}\left[\|Y_{t+1} - \hat{Y}_t^1\| \leq C_t^1\right] \\
&+ \mathbb{P}\left[c(\boldsymbol{x}_{t+1}, Y_{t+1}) \geq 0 \,\middle|\, \|Y_{t+1} - \hat{Y}_t^1\| > C_t^1\right] \mathbb{P}\left[\|Y_{t+1} - \hat{Y}_t^1\| > C_t^1\right] \\
&\overset{(b)}{\geq} \mathbb{P}\left[c(\boldsymbol{x}_{t+1}, Y_{t+1}) \geq 0 \,\middle|\, \|Y_{t+1} - \hat{Y}_t^1\| \leq C_t^1\right] \mathbb{P}\left[\|Y_{t+1} - \hat{Y}_t^1\| \leq C_t^1\right] \\
&\overset{(c)}{=} \mathbb{P}\left[\|Y_{t+1} - \hat{Y}_t^1\| \leq C_t^1\right],
\end{aligned}$$

where the equality in (a) follows from the law of total probability, while the inequality in (b) follows from the nonnegativity of probabilities. The equality in (c) follows as

$$\mathbb{P}[c(\boldsymbol{x}_{t+1}, Y_{t+1}) \geq 0 \text{ s.t. } \|Y_{t+1} - \hat{Y}_t^1\| \leq C_t^1] = 1$$

since $\|Y_{t+1} - \hat{Y}_t^1\| \leq C_t^1$ implies $c(\boldsymbol{x}_{t+1}, Y_{t+1}) \geq 0$ according to (6.8). The result from Corollary 8 completes the proof. □

6.4 Results and Discussion

The simulation-based empirical performance of the MPC with ACP uncertainty prediction regions is compared with prior work that uses a distributionally robust approach to uncertainty quantification (see Chapter 5). The same multirotor operating in the presence of a moving obstacle example is studied with a MPC planner. The multirotor is constrained to operate within the state constraints $\theta \in [-0.45, 0.45]$ radians and $\varphi \in [-0.45, 0.45]$ radians. The following standard multirotor linear dynamics is used,

$$\ddot{x} = -g\theta,\ \ddot{y} = g\varphi,\ \ddot{z} = u_1 - g,\ \ddot{\varphi} = \frac{u_2}{I_{xx}},\ \ddot{\theta} = \frac{u_3}{I_{yy}},\ \ddot{\psi} = \frac{u_4}{I_{zz}}, \tag{6.9}$$

, where the planner control inputs u_1, u_2, u_3, u_4, corresponds to the thrust force in the body frame and three moments. The vehicle's moments of inertia are $I_{xx} = 0.0075 kgm^2, I_{yy} = 0.0075 kgm^2, I_{zz} = 0.013 kgm^2$. The MPC planner has a horizon length of 10 steps and the planner is updated at 20 Hz. It is implemented through a Sequential Convex Programming approach [90].

Numerical simulations of the proposed MPC planner with ACP regions and dynamics (6.9) are presented as it avoids a Frisbee that is thrown at the drone from various initial positions, velocities, and rotation speeds. The Frisbee is modeled following [167], and linear predictions of the trajectory arising from its nonlinear dynamics are considered.

A total of 1000 Monte Carlo simulations are conducted per allowed failure probability level ϵ to compare the numerical feasibility, percentage of success in obstacle avoidance (if the MPC planner is feasible), and the planner's conservativeness, as measured by the minimum distance between the obstacle and agent centers, *i.e.*, $\bar{d}_{min}$ and $\sigma(d_{min})$) describe the average and standard deviation of this minimum distance across simulations, respectively. Three uncertainty quantification techniques are compared and tabulated in Table 5.1, (1) The proposed ACP method (Algorithm 5), (2) empirical bootstrap prediction that accounts for the prediction uncertainty using the empirical bootstrap variance (see Chapter 5), and (3) the sliding linear predictor with an Extended Kalman Filter (EKF) that provides Gaussian approximations of the obstacle prediction uncertainty (see Chapter 7).

Table 5.1 shows that the proposed method can successfully avoid the Frisbee while using a significantly smaller average divergence distance (d_{min}, $\sigma(d_{min})$) from the Frisbee. To wit, the proposed approach avoids the conservatism of other approaches due to the adaptivity of the uncertainty sets. The proposed method can usefully

Case	δ	0.025		
	UQ method	Proposed	[15]	w/EKF
Frisbee w/drag	%Feas.	83.8	87.4	97.1
	%Succ.	99.2	100	100
	$\overline{d}_{min}$	2.91	14.2	5.27
	$\sigma(d_{min})$	1.25	2.04	1.28

Table 6.1: Summary of results from MC simulations of system (6.9). FACP is implemented for predicting uncertainty sets with learning rates $\gamma = \{0.0008, 0.0015, 0.003, 0.005, 0.009, 0.017, 0.03, 0.05, 0.08, 0.13\}$ and using the last 30 measurements of the obstacle.

Case	δ	0.05		
	UQ method	Proposed	[15]	w/EKF
Frisbee w/drag	%Feas.	80.9	90.3	97.6
	%Succ.	100	100	100
	$\overline{d}_{min}$	2.74	4.97	4.25
	$\sigma(d_{min})$	1.3	1.97	1.11

Table 6.2: Summary of results from MC simulations of system (6.9). FACP is implemented for predicting uncertainty sets with learning rates $\gamma = \{0.0008, 0.0015, 0.003, 0.005, 0.009, 0.017, 0.03, 0.05, 0.08, 0.13\}$ and using the last 30 measurements of the obstacle.

adjust the prediction sets when the underlying uncertainty distribution is shifting (due to discrepancy in the linear dynamic predicted and the true nonlinear obstacle motion). It is important to note that the feasibility of the MPC optimization is worse for this approach compared to the results using the bootstrap distributionally robust formulation in Chapter 5) and the Koopman-inspired predictor with only heuristic unsafe set in Chapter 7. This issue arises during sudden changes in the size of the uncertainty sets when the learning rate γ is chosen too large. This issue is considered as future direction by considering tools to ensure recursive feasibility [178] or by providing backup controllers [179], [180] when the MPC is infeasible.

This chapter presents an algorithm for safe motion planning in an environment with other dynamic agents using ACP. Specifically, this chapter investigates a deterministic control system that uses state predictors to estimate the future motion of dynamic agents. Leveraging ideas from ACP, the prediction uncertainty is dynamically quantified from an online data stream. An uncertainty-informed model predictive controller is proposed to safely navigate among dynamic agents incorporating the quantified prediction uncertainties. In contrast to other data-driven prediction models that quantify prediction uncertainty in a heuristic manner, the proposed method produces true prediction uncertainty in a distribution-free, adaptive manner that even allows capturing changes in prediction quality and the agents’ motion.

Chapter 7

MOA: PROBABILISTIC SAFETY GUARANTEES

The growing deployment of robots in urban environments has significantly amplified the need for effective online robotic path planning and control strategies, particularly in the context of dynamic obstacle avoidance [45], [46]. Applications include autonomous driving in the presence of other vehicles [181] and pedestrians [182], autonomous drone racing while maintaining a safe distance from other competitors [183], and air traffic control where the imperatives of fuel economy, order, and safety are enforced [184]. In such unpredictable and complex environments, quantifying uncertainties becomes paramount for ensuring safety in motion planning frameworks.

Moving obstacle avoidance in a real-world setting is further complicated by hardware constraints such as limited-time position observations, sensor data at low frequency, and noise in measurements. Achieving obstacle avoidance guarantees usually requires an understanding of the obstacles' behavior, in order to define collision-free safe regions [27], [55], [185], [186]. Koopman operators can learn a lifted linear representation of nonlinear dynamical systems from data to make obstacle trajectory predictions [187], [188]. Typically, making obstacle trajectory predictions from a learned model provides limited safety information, requiring rigorous prediction uncertainty quantification to establish guarantees [189]. However, constructing deterministic robust safe regions that guarantee collision-free movements can lead to over-conservatism [27], [186]. As a result, probabilistic safe regions have gained favor, as they offer a balance between reducing conservatism and infeasibility while offering tunability based on risk preferences [15], [177].

This chapter aims to establish a comprehensive framework for moving obstacle avoidance, particularly in situations where the dynamics and intentions of obstacles are unknown and their measurements are noisy. The proposed planning framework is not only tunable and free from heuristic approaches but also provides provable safety guarantees. The core of the approach is to estimate collision-free regions, relying solely on real-time measurements of obstacle positions, and efficiently recast these prediction regions into a convex and deterministic finite-time optimal control problem that has probabilistic safety guarantees.

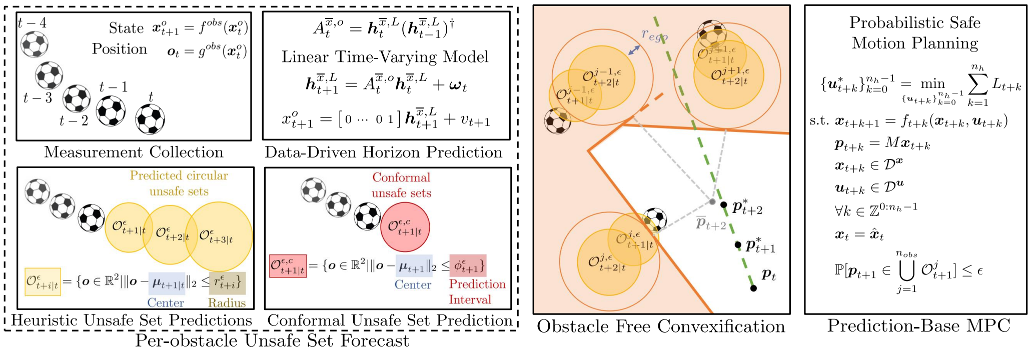
$t-4$
State $\boldsymbol{x}^o_{t+1} = f^{obs}(\boldsymbol{x}^o_t)$
Position $\boldsymbol{o}_t = g^{obs}(\boldsymbol{x}^o_t)$
$t-3$
$t-2$
$t-1$
t
Measurement Collection
$A^{\overline{x},o}_t = \boldsymbol{h}^{\overline{x},L}_t(\boldsymbol{h}^{\overline{x},L}_{t-1})^\dagger$
Linear Time-Varying Model
$\boldsymbol{h}^{\overline{x},L}_{t+1} = A^{\overline{x},o}_t\boldsymbol{h}^{\overline{x},L}_t + \boldsymbol{\omega}_t$
$x^o_{t+1} = [0 \cdots 0\ 1]\,\boldsymbol{h}^{\overline{x},L}_{t+1} + v_{t+1}$
Data-Driven Horizon Prediction
Predicted circular unsafe sets
$\mathcal{O}^\epsilon_{t+1|t}$
$\mathcal{O}^\epsilon_{t+2|t}$
$\mathcal{O}^\epsilon_{t+3|t}$
$\mathcal{O}^\epsilon_{t+i|t} = \{\boldsymbol{o} \in \mathbb{R}^2 | \|\boldsymbol{o} - \boldsymbol{\mu}_{t+1|t}\|_2 \leq r^\epsilon_{t+i}\}$
Center
Radius
Heuristic Unsafe Set Predictions
Conformal unsafe sets
$\mathcal{O}^{\epsilon,c}_{t+1|t}$
$\mathcal{O}^{\epsilon,c}_{t+1|t} = \{\boldsymbol{o} \in \mathbb{R}^2 | \|\boldsymbol{o} - \boldsymbol{\mu}_{t+1}\|_2 \leq \phi^\epsilon_{t+1}\}$
Center
Prediction Interval
Conformal Unsafe Set Prediction
Per-obstacle Unsafe Set Forecast
r_{ego}
$\mathcal{O}^{j-1,\epsilon}_{t+2|t}$
$\mathcal{O}^{j-1,\epsilon}_{t+1|t}$
$\mathcal{O}^{j+1,\epsilon}_{t+1|t}$
$\mathcal{O}^{j+1,\epsilon}_{t+2|t}$
$\boldsymbol{p}^*_{t+2}$
$\overline{p}_{t+2}$
$\mathcal{O}^{j,\epsilon}_{t+1|t}$
$\mathcal{O}^{j,\epsilon}_{t+2|t}$
$\boldsymbol{p}^*_{t+1}$
$\boldsymbol{p}_t$
Obstacle Free Convexification
Probabilistic Safe Motion Planning
$\{\boldsymbol{u}^*_{t+k}\}_{k=0}^{n_h-1} = \min_{\{\boldsymbol{u}_{t+k}\}_{k=0}^{n_h-1}} \sum_{k=1}^{n_h} L_{t+k}$
s.t. $\boldsymbol{x}_{t+k+1} = f_{t+k}(\boldsymbol{x}_{t+k}, \boldsymbol{u}_{t+k})$
$\boldsymbol{p}_{t+k} = M\boldsymbol{x}_{t+k}$
$\boldsymbol{x}_{t+k} \in \mathcal{D}^{\boldsymbol{x}}$
$\boldsymbol{u}_{t+k} \in \mathcal{D}^{\boldsymbol{u}}$
$\forall k \in \mathbb{Z}^{0:n_h-1}$
$\boldsymbol{x}_t = \hat{\boldsymbol{x}}_t$
$\mathbb{P}[\boldsymbol{p}_{t+1} \in \bigcup_{j=1}^{n_{obs}} \mathcal{O}^j_{t+1}] \leq \epsilon$
Prediction-Base MPC

Figure 7.1: Overall architecture of the proposed framework from data-driven obstacle model extraction, trajectory forecast, and prediction uncertainty quantification to constructing collision-free convex state constraints which are incorporated in an MPC to plan collision-avoiding actions upholding a user-specified risk-tolerance.

The proposed framework is depicted in Fig. 7.1, consists of three main components: (1) a Data-Driven Horizon Predictor, (2) Heuristic and Conformal Unsafe Sets, and (3) a Prediction-Aware Planner. This framework centers around an optimization-based model predictive planner that tactically incorporates deterministic predicted unsafe sets constructed from rigorous uncertainty quantification.

This chapter was adapted from: [16]

7.1 Problem Statement

This section reviews a general dynamic description of the ego vehicle. This modeling choice can capture general mobile-based systems while keeping analytical tractability. Consider the following discrete-time linear dynamical system:

$$\boldsymbol{x}_{t+1} = A_t\boldsymbol{x}_t + B_t\boldsymbol{u}_t + \boldsymbol{c}_t \tag{7.1}$$

$$\boldsymbol{p}_t = M\boldsymbol{x}_t \tag{7.2}$$

where $\boldsymbol{x}_t \in \mathcal{D}^{\boldsymbol{x}} \subset \mathbb{R}^{n_{\boldsymbol{x}}}$ is the state of the system and $\boldsymbol{u}_t \in \mathcal{D}^{\boldsymbol{u}} \subset \mathbb{R}^{n_{\boldsymbol{u}}}$ denotes the control input, and $\mathcal{D}^{\boldsymbol{x}}$ and $\mathcal{D}^{\boldsymbol{u}}$ represents the admissible convex state and input sets. The time-varying matrices at time t, $A_t \in \mathbb{R}^{n_{\boldsymbol{x}} \times n_{\boldsymbol{x}}}$, $B_t \in \mathbb{R}^{n_{\boldsymbol{x}} \times n_{\boldsymbol{u}}}$, and $\boldsymbol{c}_t \in \mathbb{R}^{n_{\boldsymbol{x}}}$ are the state transition matrix, actuation matrix, and time-varying additive model residuals (known), respectively. The position vector, $\boldsymbol{p} \in \mathbb{R}^3$, is the system output in 3D Euclidean space relative to an inertial coordinate. Let the position output of the vehicle be mapped from the system state $\boldsymbol{x}_t$ by a constant (time-invariant) observation matrix $M \in \mathbb{R}^{3 \times n_{\boldsymbol{x}}}$. For simplicity, the ego vehicle is assumed to be full-state observable. The ego vehicle model is (7.1)-(7.2) which is a general nonlinear continuous-time system with dynamics $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$ can be reformulated to (7.1)-(7.2) by linearization and discretization.

Specifically, the matrices A_t, B_t, and $\boldsymbol{c}_t$ in (7.1) can be obtained from linearizing and taking the first-order linearization and discretization of the dynamics as:

$$A_t \triangleq I_{n_x} + \Delta T \frac{\partial f}{\partial \boldsymbol{x}_t}\bigg|_{(\overline{\boldsymbol{x}}_t, \overline{\boldsymbol{u}}_t)}, B_t \triangleq \Delta T \frac{\partial f}{\partial \boldsymbol{u}_t}\bigg|_{(\overline{\boldsymbol{x}}_t, \overline{\boldsymbol{u}}_t)}, \tag{7.3}$$

$$\boldsymbol{c}_t \triangleq f(\overline{\boldsymbol{x}}_t, \overline{\boldsymbol{u}}_t) - A_t \overline{\boldsymbol{x}}_t - B_t \overline{\boldsymbol{u}}_t. \tag{7.4}$$

Here, $\overline{\boldsymbol{x}}_t$ and $\overline{\boldsymbol{u}}_t$ denote the nominal state and input trajectories which the nonlinear dynamics f is linearized around. The parameter ΔT is the time discretization step size.

Background

This chapter studies the case where the ego vehicle (7.1)-(7.2) is tasked to track a time-based reference position trajectory $\boldsymbol{p}^{\text{ref}}(t)$. While following this path, the ego vehicle may encounter n^{obs} spherical, static, or dynamic obstacles whose trajectories and dynamics are *a priori* unknown. The obstacles occupy $\mathcal{O}^{obs} = \bigcup_{i=0}^{n_{obs}} \mathcal{O}_j^{obs} \subset \mathbb{R}^3$, which is considered as an *unsafe* set for the ego vehicle. Dynamic obstacles are categorized into three types based on their interaction with the ego vehicle:

Definition 11. *A* ***cooperative*** *obstacle actively maintains a safe distance from the agent. An* ***indifferent*** *obstacle follows a predefined trajectory without reacting to the agent's behavior.* [1] *A* ***hostile*** *obstacle actively pursues the ego vehicle for a finite period.*

The obstacles are generalized as a dynamical system that each follows a trajectory that is governed by the following unknown discrete-time nonlinear dynamics:

$$\begin{bmatrix} \boldsymbol{o}_{t+1} \\ \boldsymbol{\zeta}_{t+1} \end{bmatrix} = f_t^{obs}\left(\boldsymbol{o}_t, \boldsymbol{\zeta}_t\right) \tag{7.5}$$

where $\boldsymbol{o}_t \triangleq [x_t^o, y_t^o, z_t^o]^T \in \mathbb{R}^3$ represents the x, y, z position of the obstacle in the inertial frame, and $\boldsymbol{\zeta}_t \in \mathbb{R}^{n_o - 3}$, integer $n_o \geq 3$, denotes the unobservable obstacle states and/or inputs which may or may not include the ego vehicle's states. These unobservable states may include unmeasurable states such as the spinning rate of a Frisbee that plays an important role in the Frisbee's gilding trajectory. Nevertheless,

[1] Static obstacles are a subset of indifferent obstacles.

the function f_t^{obs}, characterizing the dynamics at time t, is assumed to be locally Lipschitz continuous with a single fixed point.

Each obstacle is modeled by its bounding sphere. The j^{th} obstacle occupies the set $O_t^j(\boldsymbol{c}_t, r) = \{\boldsymbol{o} \in \mathbb{R}^3 \mid \|\boldsymbol{c}_t - \boldsymbol{o}\|_2 \leq r\}$, where $\boldsymbol{c}_t \in \mathbb{R}^3$ and $r \in \mathbb{R}_{>0}$ represent the geometric center and constant radius of the j^{th} obstacle at time t, respectively. The objective of this work is to determine an optimal control input, $\boldsymbol{u}_t^*$, at any given time t. This control input should enable the ego vehicle to track the reference position, denoted as $\boldsymbol{p}_{t+1}^{ref}$, as closely as possible given its current state $\boldsymbol{x}_t$. Simultaneously, it must comply with state and input constraints, as well as ensure avoidance of any obstacles as defined above.

Problem Statement

The process of avoiding moving obstacles with unknown trajectories, dynamics, and intent can often lead to overly cautious behaviors. Researchers have tackled this challenge by predicting obstacle trajectories or classifying obstacle behaviors to plan avoidance trajectories [190]. Various efforts attempted to predict obstacle behaviors as probabilistic distributions [74], [191]. Incorporating the quantified prediction model uncertainty, this method outputs a probabilistic distribution of the obstacle position.

The prediction sets, recognized as unsafe sets, can be incorporated into the model predictive planner as constraints [14], [15], [80] or as artificial potential fields [192], [193] for generating trajectories. Ideally, these prediction sets should reflect a risk-liked metric such that the user can balance between conservatism and its application objectives. As an example, the risk of damaging the vehicle hardware is less important compared to the time it takes to localize survivors in search and rescue missions. Autonomous driving, on the contrary, may desire to minimize collision risks and be receptive to rather conservative paths to avoid pedestrians.

In summary, the objective is to obtain predictions $\{\boldsymbol{o}_{t+1|t}^j, \ldots, \boldsymbol{o}_{t+n_h|t}^j\}$ at time t of future obstacle positions $\{\boldsymbol{o}_{t+1}^j, \ldots, \boldsymbol{o}_{t+n_h}^j\}$, over the planning horizon n_h, from past observations $(\hat{\boldsymbol{o}}_0^j, \ldots, \hat{\boldsymbol{o}}_t^j)$ of the j^{th} obstacle. Specifically, the goal is to obtain a model characterizing the obstacle's closed-loop behavior which can be used to make future obstacle position predictions by forward propagating this model and its quantified prediction uncertainties.

The obstacle position is assumed to be measurable at a fixed frequency, f_s, after detecting the obstacle. Using a 3D obstacle as an example, the obstacle measurements are denoted as $\hat{\boldsymbol{o}}_t^j = \boldsymbol{o}_t^j + \delta \boldsymbol{o}_t$ where $\delta \boldsymbol{o}_t \in \mathcal{N}(\mathbf{0}_{3\times1}, \Sigma_{\boldsymbol{o},j})$ and $\mathbf{0}_{3\times1} \in \mathbb{R}^3$ is a zero vector indicating the expected value (zero-mean) and $\Sigma_{\boldsymbol{o},j} = \text{diag}(\sigma_{x,j}, \sigma_{y,j}, \sigma_{z,j})$ is a diagonal covariance matrix of the multivariate normal distribution (white-noise).

Problem 5. ***[Data-Driven Obstacle Model]*** *Consider a multivariate stochastic process where observables $\{x_i^{o,j}\}_{i=0}^t$, $\{y_i^{o,j}\}_{i=0}^t$, and $\{z_i^{o,j}\}_{i=0}^t$ denote the spherical obstacle's true center location in the ego vehicle's inertial frame. The measurements are corrupted by noises $\{\delta \boldsymbol{o}_i\}_{i=0}^t$, drawn independently from a normal distribution $\mathcal{N}(\mathbf{0}_{3\times1}, \Sigma_{\boldsymbol{p},j})$ (see Fig. 7.2). The problem objective is to learn a dynamic model for the closed-loop obstacle behavior which is defined as a time update map h_t that predict obstacle position at time $t+1$ given current time t, i.e., $\tilde{\boldsymbol{o}}_{t+1} = h_t(\boldsymbol{q}_t^{obs})$ where $\boldsymbol{q}_t^{obs} \in Q_t^{obs}$ represents the inputs to the map $h_t : Q_t^{obs} \rightarrow O_t^{obs} \subseteq \mathbb{R}^3$. Specifically, the probability space $(O_t^{obs}, \mathcal{F}_t^{obs}, \mathbb{P}_t)$ with a sample space O_t^{obs}, a $\sigma-$algebra $\mathcal{F}_t$ over O_t^{obs} events, and a probability measure $\mathbb{P}_t$ describes the predictor outcomes at time t.*

Given the data-driven model, the follow-on question is to obtain probabilistic unsafe sets given a user-defined confidence/risk level to account for model uncertainties, including those driven by learning errors.

Problem 6. ***[Unsafe Set Prediction]*** *Let $h_t : Q_t \rightarrow O_t^{obs}$ at time t, be the solution to Problem 5 given predictor input $\boldsymbol{q}_t^{obs}$, as the data-driven obstacle model. The problem objective is to identify a time-varying metric $r_{t+1}^{\epsilon} \in \mathbb{R}_{>0}$ and $r_{t+1}^{\epsilon} \geq \rho_{t+1}^{\epsilon}$ given a risk-level $\epsilon \in [0,1]$ which is the radius of the risk dependent unsafe set defined as*

$$\rho_{t+1}^{\epsilon} \triangleq \min_{\rho_{t+1} \in \mathbb{R}_{>0}} \{\mathbb{P}_{t+1}[\|\boldsymbol{o}_{t+1} - \tilde{\boldsymbol{o}}_{t+1}\| \leq \rho_{t+1}] \geq 1 - \epsilon\}, \tag{7.6}$$

where $\boldsymbol{o}_{t+1}$ is the true obstacle position at time $t+1$ and $\tilde{\boldsymbol{o}}_{t+1} \in O_{t+1}^{obs}$ is the predicted obstacle position at time $t+1$ made at time t using the obstacle model h_t. Denote the $1-\epsilon$ confidence unsafe set prediction at time t as

$$O_{t+1}^{\epsilon} \triangleq \{\boldsymbol{o} \in \mathbb{R}^3 \mid \|\boldsymbol{o}_{t+1} - \tilde{\boldsymbol{o}}_{t+1}\| \leq r_{t+1}^{\epsilon}\} \tag{7.7}$$

where $r_{t+1}^{\epsilon} \geq \rho_{t+1}^{\epsilon}$ is the predicted unsafe radius.

The obstacle position predictions, denoted as $\tilde{\boldsymbol{o}}_{t+1}$, are the center of the unsafe set. Ideally, the tightest unsafe radius ρ_{t+1}^{ϵ} is desirable, given the quantified uncertainty, to avoid being overly conservative. Due to limited and noisy partial data and the lack of explicit dynamics models, the tightness guarantees can only be obtained with additional assumptions.

Problem 7. ***[Probabilistic Obstacle Avoidance]*** *Consider the dynamical system* (7.1)-(7.2) *with convex state constraint $\mathcal{D}^{\boldsymbol{x}}$ and input constraint $\mathcal{D}^{\boldsymbol{u}}$. A desired agent position $\boldsymbol{p}_{t+1}^{ref}$ is given current at time t. Suppose there exists a spherical obstacle, whose the true occupied unsafe set is $\mathcal{O}_{t+1}$. The planner must compute a control input $\boldsymbol{u}_t^{safe} \in \mathcal{D}^{\boldsymbol{u}}$ that guarantees obstacle avoidance at with least $(1-\epsilon)$ probability at time $t+1$. This goal can be compactly formulated as the following constraint*

$$\mathbb{P}_{t+1}\left[\mathcal{O}_{t+1} \subseteq \mathcal{O}_{t+1}^{\epsilon}\right] \geq 1-\epsilon, \tag{7.8}$$

The set $\mathcal{O}_{t+1}^{\epsilon}$ is the $(1-\epsilon)$ confidence unsafe set prediction of the obstacle at time $t+1$, the result of Problem 6. If constraint (7.8) *is satisfied for all $t \in \mathbb{R}_{>0}$, the planning algorithm is guaranteed to achieve* ***probabilistic obstacle avoidance****. As an extension to finite T, defining* ***probabilistic obstacle avoidance on average****, a time-averaged obstacle avoidance behavior satisfied for all t, mathematically described by following constraint,*

$$\lim_{T\to\infty}\left(\frac{1}{T}\sum_{i=t-T+1}^{t}\mathbb{P}_{i+1}[\boldsymbol{p}_{i+1} \notin \mathcal{O}_{i+1}]\right) \geq 1-\epsilon, \tag{7.9}$$

where $T < t$ is a time-averaging window. However, the probabilistic avoidance on average implicitly assumes $t \to \infty$ which can be impractical to assert for real-time obstacle avoidance applications. Alternatively, an algorithm achieves ***finite-time probabilistic obstacle avoidance on average*** *if the following holds for a finite T*

$$\frac{1}{T}\sum_{i=t-T+1}^{t}\mathbb{P}_{i+1}[\boldsymbol{p}_{i+1} \notin \mathcal{O}_{i+1}] \geq 1-\epsilon. \tag{7.10}$$

While probabilistic obstacle avoidance on average provides weaker safety guarantees, it is worth highlighting that the proposed framework can provide provable probabilistic guarantees for the likelihood of the ego vehicle avoiding an obstacle without making any assumptions about the obstacle's dynamics or intentions, relying solely on noisy position measurements. In addition, the probability statements (7.8),(7.9), and (7.10) can be extended in the presence of multiple moving obstacles where the true obstacles occupied unsafe set $\mathcal{O}_{t+1}$ and predicted unsafe set $\mathcal{O}_{i+1}^{\epsilon}$ became the unions of the respective sets for all obstacles.

Lastly, a motion planner is constructed that can generate safe maneuvers for the ego vehicle while satisfying the identified probabilistic obstacle avoidance constraints, the result of Problem 7.

Problem 8. ***[Probabilistic Safe Motion Planning]*** *Let the system dynamics be* (7.1)-(7.2). *Suppose there exists a discrete-time position reference trajectory* $\boldsymbol{p}_t^{ref}$, $\forall t \in \mathbb{Z}_1^{n_h}$ *where* $n_h \in \mathbb{Z}_{>0}$ *is the length of the horizon, convex state constraints* $\mathcal{D}^{\boldsymbol{x}} \subseteq \mathbb{R}^{n_x}$*, convex input constraints* $\mathcal{D}^{\boldsymbol{u}} \subseteq \mathbb{R}^{n_u}$*, and a convex stage cost* $\mathcal{L}_i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \longrightarrow \mathbb{R}_{\geq 0}$ *of the form*

$$\mathcal{L}_i = (\boldsymbol{p}_i^{ref} - \boldsymbol{p}_i)^T Q (\boldsymbol{p}_i^{ref} - \boldsymbol{p}_i) + \boldsymbol{u}_i^T R \boldsymbol{u}_i,$$

where matrices Q *and* R *are positive-definite. Suppose a total of* n_{obs} *spherical obstacles are present in the environment and an ego vehicle has current position* $\boldsymbol{p}_t$*. Given a risk tolerance* $\epsilon \in (0, 1]$*, the problem objective is to compute a receding horizon controller* $\{\boldsymbol{u}^*\}_1^{n_h}$ *that minimizes* $\sum_{i=1}^{n_h} \mathcal{L}_{t+i}$ *while the satisfying state, input, and probabilistic obstacle avoidance on average in finite time* (7.10).

7.2 Horizon Trajectory Predictor

Let noisy observable obstacle positions be $\{\hat{\boldsymbol{o}}_k^j\}_{k=0}^t$, where the subscript j is the obstacle index and the subscript k is the time time index. [2] The objective, viewed from the perspective of the Koopman operator, is to derive a linear model that characterizes the evolution of obstacle positions over time. To achieve this, a variant dynamic mode decomposition (DMD) is proposed to numerically extract the obstacle model. See Fig. 7.2 as an overview of the horizon predictor.

Time-delay Embedding

In theory, with ample data, computational resources, and knowledge of true eigenfunctions, a lifted Koopman operator $A^o \in \mathbb{R}^{\infty \times \infty}$ can be calculated for any obstacle that is time-invariant and describes the time-evolution of the system observables. However, it will be both unattainable and unpractical in a real-time obstacle avoidance setting. Instead, a finite-dimensional approximation of $A_t^o \in \mathbb{R}^{n_l \times n_l}$ is desriable, where $n_l << \infty$, which might be time-varying (time indicated by its subscript t) linear matrix due to finite-rank truncation. Specifically, the following linear time-varying input-output model with additive process and measurement noises is employed to describe the unknown obstacle closed-loop dynamic:

$$\boldsymbol{h}_{t+1} = A_t^o \boldsymbol{h}_t + \boldsymbol{w}_t^o, \tag{7.11}$$

$$\boldsymbol{o}_t = C^o \boldsymbol{h}_t + \boldsymbol{v}_t^o, \tag{7.12}$$

where $\boldsymbol{h}_t \in \mathbb{R}^{n_l}$ represents the lifted states of the obstacle at time t, the additive error $\boldsymbol{\omega}_t^o \in \mathbb{R}^{n_l}$ is a model discrepancy from finite-dimensional truncation, the matrix

[2] For the rest of this section, the superscript j will be dropped since the horizon trajectory predictor is applied to each obstacles separately.

$C^o \in \mathbb{R}^{3\times n_l}$ is some constant projection matrix that maps the lifted states to the obstacle position's current position, and $\boldsymbol{\nu}_t^o \in \mathbb{R}^3$ represents a time-varying additive measurement noise in the obstacle position. To be explicit, the following assumption is made regarding the time-delay embedded system behavior.

Assumption 7. *The time-varying propagator $A_t^o \in \mathbb{R}^{L\times L}$ for a time delay-embedded lifted state $\boldsymbol{h} \in \mathbb{R}^L$ satisfying $\boldsymbol{h}_{t+1} = A_t^o \boldsymbol{h}_t + \boldsymbol{\omega}_t$ has constant rank $r \leq L$ for all t. Furthermore, the rank of $\prod_{i=0}^t A_k^o$ is r for all $t \to \infty$.*

Without assuming the knowledge of r, by picking a sufficiently large L, it is critical to identify r such that key spatial-temporal modes are captured.

Time-delay embedding, a classical technique [66] for dealing with partial state information, is applied to the obstacle position measurements to construct the lifted states $\boldsymbol{h}_t$ from position observables. Recent connections between time-delay embedding and the Koopman frameworks [100], [187], [194]–[196] have revealed that the time-delay embedding representation of state trajectories can serve as a universal eigenfunction for constructing Koopman operators for autonomous systems. Furthermore, the spectral decomposition of time-delay coordinates has been explored in other fields, such as Eigensystem Realization Algorithms ([197]) and Singular Spectrum Analysis ([87], [198]).

Hankel Embedding

Introducing a L-delay embedding vector at time t using the obstacle x positions using the most recent L observables $\boldsymbol{h}_t^{\hat{x},L} \triangleq [\hat{x}_{t-L+1}, \hat{x}_{t-L+2}, \cdots, \hat{x}_t]^T \in \mathbb{R}^L$. The superscript identifier $\hat{x}$ denotes that the noisy x observables are used for constructing the delay embedding vector. Superscript L denotes the embedding time length or window. Subscript t denotes the time instance of the most recent observable used in the construction of the embedding vector. By stacking time-delay embedded vectors horizontally with increasing t, the x position L-embedding Hankel matrix $H_t^{\hat{x},L,N} \in \mathbb{R}^{L\times N}$ can be similarly constructed using the most recent N measurements, where $N \geq L$, as:

$$H_t^{\hat{x},L,N} = \left[\boldsymbol{h}_{t-N+1}^{\hat{x},L} \quad \boldsymbol{h}_{t-N+2}^{\hat{x},L} \quad \cdots \quad \boldsymbol{h}_t^{\hat{x},L}. \right]. \tag{7.13}$$

Repeating patterns in the Hankel matrix represent underlying trends and oscillations which can be extracted from performing Singular Value Decomposition, $H_t^{\hat{x},L,N} = U\Sigma V^*$. Matrix $U = \left[\boldsymbol{\mu}_1^H, \boldsymbol{\mu}_H^2, \ldots, \boldsymbol{\mu}_L^H\right] \in \mathbb{R}^{L\times L}$, is a unitary matrix

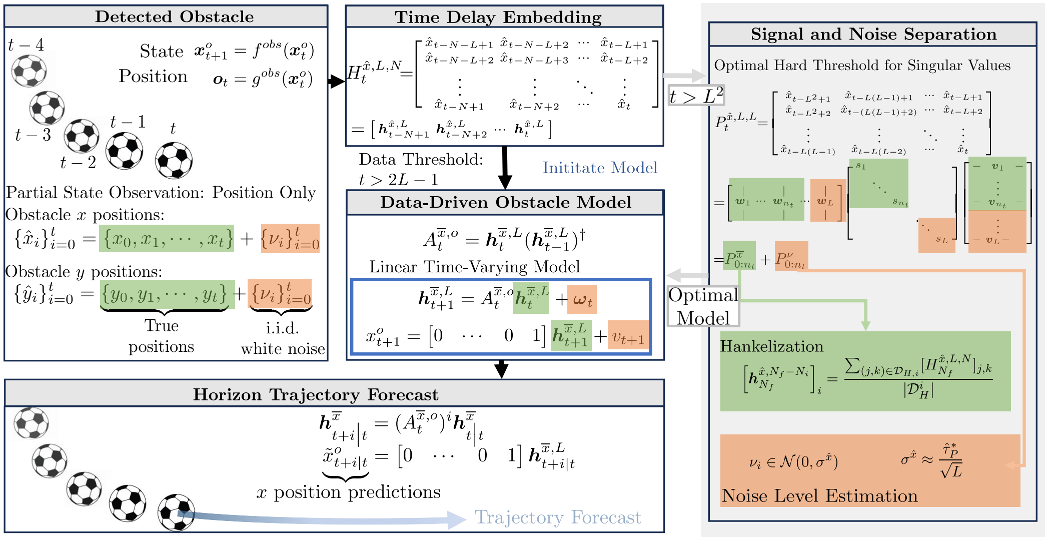
Detected Obstacle
$t-4$
State $\boldsymbol{x}^o_{t+1} = f^{obs}(\boldsymbol{x}^o_t)$
Position $\boldsymbol{o}_t = g^{obs}(\boldsymbol{x}^o_t)$
$t-3$
$t-2$
$t-1$
t
Partial State Observation: Position Only
Obstacle x positions:
$\{\hat{x}_i\}_{i=0}^t = \{x_0, x_1, \cdots, x_t\} + \{\nu_i\}_{i=0}^t$
Obstacle y positions:
$\{\hat{y}_i\}_{i=0}^t = \underbrace{\{y_0, y_1, \cdots, y_t\}}_{\text{True positions}} + \underbrace{\{\nu_i\}_{i=0}^t}_{\text{i.i.d. white noise}}$
Time Delay Embedding
$H_t^{\hat{x},L,N} = \begin{bmatrix} \hat{x}_{t-N-L+1} & \hat{x}_{t-N-L+2} & \cdots & \hat{x}_{t-L+1} \\ \hat{x}_{t-N-L+2} & \hat{x}_{t-N-L+3} & \cdots & \hat{x}_{t-L+2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{x}_{t-N+1} & \hat{x}_{t-N+2} & \cdots & \hat{x}_t \end{bmatrix}$
$= \left[\boldsymbol{h}_{t-N+1}^{\hat{x},L} \; \boldsymbol{h}_{t-N+2}^{\hat{x},L} \cdots \boldsymbol{h}_t^{\hat{x},L} \right]$
Data Threshold:
$t > 2L-1$
Inititate Model
Data-Driven Obstacle Model
$A_t^{\overline{x},o} = \boldsymbol{h}_t^{\overline{x},L}(\boldsymbol{h}_{t-1}^{\overline{x},L})^\dagger$
Linear Time-Varying Model
$\boldsymbol{h}_{t+1}^{\overline{x},L} = A_t^{\overline{x},o}\boldsymbol{h}_t^{\overline{x},L} + \boldsymbol{\omega}_t$
$x_{t+1}^o = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} \boldsymbol{h}_{t+1}^{\overline{x},L} + v_{t+1}$
$t > L^2$
Optimal Model
Signal and Noise Separation
Optimal Hard Threshold for Singular Values
$P_t^{\hat{x},L,L} = \begin{bmatrix} \hat{x}_{t-L^2+1} & \hat{x}_{t-L(L-1)+1} & \cdots & \hat{x}_{t-L+1} \\ \hat{x}_{t-L^2+2} & \hat{x}_{t-(L(L-1)+2)} & \cdots & \hat{x}_{t-L+2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{x}_{t-L(L-1)} & \hat{x}_{t-L(L-2)} & \cdots & \hat{x}_t \end{bmatrix}$
$= \begin{bmatrix} \boldsymbol{w}_1 & \cdots & \boldsymbol{w}_{n_t} & \cdots & \boldsymbol{w}_L \end{bmatrix} \begin{bmatrix} s_1 & & & & \\ & \ddots & & & \\ & & s_{n_t} & & \\ & & & \ddots & \\ & & & & s_L \end{bmatrix} \begin{bmatrix} - \boldsymbol{v}_1 - \\ \vdots \\ - \boldsymbol{v}_{n_t} - \\ \vdots \\ - \boldsymbol{v}_L - \end{bmatrix}$
$= P_{0:n_l}^{\overline{x}} + P_{0:n_l}^{\nu}$
Hankelization
$\left[\boldsymbol{h}_{N_f}^{\hat{x},N_f-N_i}\right]_i = \frac{\sum_{(j,k)\in\mathcal{D}_{H,i}}[H_{N_f}^{\hat{x},L,N}]_{j,k}}{|\mathcal{D}_H^i|}$
$\nu_i \in \mathcal{N}(0,\sigma^{\hat{x}})$
$\sigma^{\hat{x}} \approx \frac{\hat{\tau}_P^*}{\sqrt{L}}$
Noise Level Estimation
Horizon Trajectory Forecast
$\boldsymbol{h}_{t+i|t}^{\overline{x}} = (A_t^{\overline{x},o})^i \boldsymbol{h}_{t|t}^{\overline{x}}$
$\underbrace{\tilde{x}_{t+i|t}^o}_{x \text{ position predictions}} = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} \boldsymbol{h}_{t+i|t}^{\overline{x},L}$
Trajectory Forecast

Figure 7.2: Illustration of the proposed data-driven horizon predictor in Section 7.2 for intention unknown dynamic obstacles. With only noisy past position measurements, a linear time-varying model is extracted online for making obstacle position forecasts. The data-driven obstacle model will improve in fidelity and performance as more obstacle measurements are accumulated. Reaching data threshold $t_1 = 2L - 1$ allows minimum time-delay embedding and model reduction. After reaching threshold $t_2 = L^2$, optimal signal and noise separation are available for higher fidelity model reduction and accuracy forecasts.

comprises of the spatial empirical orthogonal functions of the Hankel matrix, and $V = \left[\boldsymbol{v}_H^1, \boldsymbol{v}_H^2, \ldots, \boldsymbol{v}_H^N\right] \in \mathbb{R}^{N\times N}$ is comprised of the temporal modes. The magnitude of the singular values $\sigma_i \in \mathbb{R}$, for $i \in \mathbb{Z}_0^{L-1}$, indicates the magnitude of the spatial-temporal mode $\boldsymbol{\mu}_i^H \boldsymbol{v}_i^T$.

Definition 12. *The dehankel function $D_H : \mathbb{R}^{L\times N} \times \mathbb{Z}_+ \times \mathbb{Z}_+ \longrightarrow \mathbb{R}^{N_f - N_i}$ is a map that transforms time-delay embedded Hankel matrices into the original time signal from time index $N_i \in \mathbb{Z}_+$ to $N_f \in \mathbb{Z}_+$ where $0 \le N_i < N_f \le t$. Mathematically, the i^{th} element of the $\boldsymbol{h}_{N_f}^{\hat{x},N_f-N_i} = D_H(H_t^{\hat{x},L,N}, N_i, N_f)$ is defined as:*

$$\left[\boldsymbol{h}_{N_f}^{\hat{x},N_f-N_i}\right]_i = \frac{\sum_{(j,k)\in\mathcal{D}_{H,i}}[H_{N_f}^{\hat{x},L,N}]_{j,k}}{|\mathcal{D}_H^i|} \tag{7.14}$$

where the index set $\mathcal{D}_H^i \triangleq \{(l,n) : l + n = |\mathcal{D}_H^i|, l \in \mathbb{Z}_0^{L-1}, n \in \mathbb{Z}_0^{N-1}\}$ is the column and row index corresponds to the i^{th} secondary diagonals of the operated Hankel matrix, and $|\mathcal{D}_H^i|$ denotes the number of elements in set $\mathcal{D}_H^i$.

If a matrix has minor variations in its secondary diagonals (not an exact Hankel matrix), the dehankel function retrieves the time series signal in which its constructed

$L-$embedding Hankel matrix is closest to $[H_t^{\hat{x},L,N}]_{j,k}$ with respect to the Frobenius norm. The proof can be found in [87].

As more obstacle position measurements are acquired over time, the same-sized Hankel matrix can be similarly constructed with the most recent N measurements denoted as $H_t^{\hat{x},L,N}$ for the x positions at time t. Hankel matrices are commonly used in system identification and signal processing to isolate time series data by frequency modes which is exploited to obtain the time-varying propagator matrix A_t^o in Eqn. (7.11).

Page Matrix and Denoising

A variant of Hankel matrices, the Page matrix is an excellent tool for noise filtering [199], which is employed in this framework to isolate the true obstacle positions from the unknown measurement noises. Unlike the Hankel matrix where each adjacent column contains $L - 1$ repeated entries, Page matrices are constructed as follows:

$$P_t^{\hat{x},L,T} = \begin{bmatrix} \boldsymbol{h}_{t-(T-1)L}^{\hat{x},L} & \boldsymbol{h}_{t-(T-2)L}^{\hat{x},L} & \cdots & \boldsymbol{h}_t^{\hat{x},L} \end{bmatrix} \in \mathbb{R}^{L\times T}$$

which is constructed with the most recent TL measurements. Suppose the most recent TL true obstacle x position measurements are available where $T \geq L$ which is used to construct to Page matrix $P_t^{x,L,T} \in \mathbb{R}^{L\times T}$. The matrix $W_t^{L,T} \in \mathbb{R}^{L\times T}$ is a noise matrix with identically and independently distributed (i.i.d.), zero-mean entries and unit variance and $\sigma^{\hat{x}} \in \mathbb{R}_{\geq 0}$ is the unknown noise level. In this case, Page matrices composed built from time series noisy measurements can be decomposed as

$$P_t^{\hat{x},L,T} = P_t^{x,L,T} + \sigma^{\hat{x}} W_t^{L,T}. \tag{7.15}$$

Since page matrices do not have repeated entries of the same time series observables, results from random matrix theory can be leveraged to separate signal from noise. Specifically, the work by Gavish and Donoho [200] provides an optimal hard threshold framework to recover low-rank matrix (in this case $P_t^{x,L,T}$) from noisy data Page matrix $P_t^{\hat{x},L,T}$). By applying Singular Value Decomposition, the measurement page matrix can be decomposed as

$$P_t^{\hat{x},L,T} = \sum_{i=1}^{L} s_i^P \boldsymbol{\mu}_i^P (\boldsymbol{\nu}_i^P)^T, \tag{7.16}$$

where $\boldsymbol{\mu}_{t,i}^P \in \mathbb{R}^L$ and $\boldsymbol{\nu}_{t,i}^P \in \mathbb{R}^T$, for $i = \{1, \cdots, L\}$ are the left and right singular vectors of $P_t^{\hat{x},L,T}$ corresponding to the singular value $s_{t,i}^P \in \mathbb{R}_{\geq 0}$ and $s_{t,i}^P \leq s_{t,i-1}^P$

for all $i \in \mathbb{Z}_2^{L-1}$. To isolate the true single matrix $P_t^{x,L,T}$, the following problem is considered: Determine the optimal SVHT value $\tau_P^* \in \{s_1^P, \cdots, s_L^P\}$ and define the τ_P-threshold Page matrix as

$$P_{t,\tau_x}^{\hat{x},L,T} = \sum_{i=1}^{L} s_i^P \mathbf{1}_{s_i^P \geq \tau_P^*} \boldsymbol{\mu}_i^P (\boldsymbol{\nu}_i^P)^T \tag{7.17}$$

that minimizes the following cost, the mean square error (MSE) relative to the true signal Page matrix

$$\begin{aligned} \tau_P^* &= \min_{\tau_x} \| P_{t,\tau_x}^{\hat{x},L,T} - P_t^{x,T,L} \|_F^2 \\ &= \min_{\tau_x} \sum_{i=1}^{\tau_x} \sum_{j=1}^{\tau_x} \left(\left(P_t^{\hat{x},T,L} \right)_{i,j} - \left(P_t^{x,T,L} \right)_{i,j} \right)^2, \end{aligned}$$

where the expression $(A)_{i,j}$ for matrix A means the $(i,j)^{th}$ entry of A. Adapting from the result in [200], for the case of $T = L$ where the Page matrix (7.15) is a square L-by-L matrix, the optimal singular value threshold τ_P^* with unknown noise level can be approximated as:

$$\hat{\tau}_P^* \approx 2.858 \cdot s_{\lfloor L/2 \rfloor}^P \tag{7.18}$$

where $s_{\lfloor L/2 \rfloor}^P$ is the median singular value of the perturbed matrix $P_t^{\hat{x},L,L}$. The unknown noise level $\sigma^{\hat{x}}$ can be also approximated from the hard singular value threshold as

$$\sigma^{\hat{x}} \approx \frac{\hat{\tau}_P^*}{\sqrt{L}}. \tag{7.19}$$

In summary, the singular value threshold $\hat{\tau}_P^*$ from (7.18) separates the Page matrix parts the desired signal from noise,

$$P_t^{\hat{x},L,L} = \underbrace{\sum_{i=1}^{n_x} s_i^P \boldsymbol{\mu}_i^P (\boldsymbol{\nu}_i^P)^T}_{\approx P_t^{x,L,L} = \text{signal}} + \underbrace{\sum_{i=n_x+1}^{L} s_i^P \boldsymbol{\mu}_i^P (\boldsymbol{\nu}_i^P)^T}_{\approx \sigma^{\hat{x}} W_t^{L,L} = \text{noise}}, \tag{7.20}$$

where n_x is the index of smallest singular value of $P_t^{\hat{x},L,L}$ that satisfies $s_{n_x} \geq \hat{\tau}_x^*$.

For the case where $T > L$ (the optimal singular value hard thresholding for non-square matrices), the approximate optimal threshold when the noise level is provided is a function of T/L. For this chapter, only the $T = L$ case is considered since it requires the least amount of data given an embedding length L to begin the denoising process.

Further, the $L-$embedding Hankel and Page matrices over the same time-series signals share the same rank. The optimal hard singular value threshold index n_x also indicated the rank of the $L-$embedding Hankel matrix constructed over the same time-series signal.

From a linear system theory point of view, Hankel and Page matrix over the same data length can be built as

$$\begin{aligned} H_t^{\hat{x},L,T} &= \begin{bmatrix} \boldsymbol{h}_{t-T+1}^{\hat{x},L} & A_{t-T+1}^{\hat{x},o}\boldsymbol{h}_{t-T+1}^{\hat{x},L} & \dots & \prod_{i=0}^{T-1} A_{t-i}^{\hat{x},o}\boldsymbol{h}_{t-T+1}^{\hat{x},L} \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} I & A_{t-T+1}^{\hat{x},o} & \dots & \prod_{i=0}^{T-1} A_{t-i}^{\hat{x},o} \end{bmatrix}}_{\triangleq L_H \in \mathbb{R}^{L\times TL}} \underbrace{\operatorname{diag}(\boldsymbol{h}_{t-T+1}^{\hat{x},L}, \dots, \boldsymbol{h}_{t-T+1}^{\hat{x},L})}_{\triangleq R_H \in \mathbb{R}^{TL\times T}}, \end{aligned}$$

$$\begin{aligned} P_t^{\hat{x},L,TL} &= \begin{bmatrix} \boldsymbol{h}_{t-(T-1)L}^{\hat{x},L} & \prod_{i=0}^{L-1} A_{t-i}^{\hat{x},o}\boldsymbol{h}_{t-(T-1)L}^{\hat{x},L} & \dots & \prod_{i=0}^{(T-1)L-1} A_{t-i}^{\hat{x},o}\boldsymbol{h}_{t-(T-1)L}^{\hat{x},L} \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} I & \prod_{i=0}^{L-1} A_{t-i}^{\hat{x},o} & \dots & \prod_{i=0}^{(T-1)L-1} A_{t-i}^{\hat{x},o} \end{bmatrix}}_{\triangleq L_P \in \mathbb{R}^{L\times TL}} \underbrace{\operatorname{diag}(\boldsymbol{h}_{t-(T-1)L}^{\hat{x},L}, \dots, \boldsymbol{h}_{t-(T-1)L}^{\hat{x},L})}_{R_P \in \mathbb{R}^{TL\times T}}. \end{aligned}$$

Under the assumption 7 with lifted state $\boldsymbol{h}_{t-T+1}^{\hat{x},L} =\in \mathbb{R}^L$ where the linear propagator rank $r \le L$, the stacked matrices L_H and L_P has rank r. By construction, R_H and R_P always has rank L when $T \ge L$. By matrix rank properties, the following conditions hold.

$$\begin{aligned} \operatorname{rank}(H_t^{\hat{x},L,T}) &= \operatorname{rank}(L_H R_H) = \operatorname{rank}(L_H) = r \\ \operatorname{rank}(P_t^{\hat{x},L,T}) &= \operatorname{rank}(L_P R_P) = \operatorname{rank}(L_P) = r. \end{aligned}$$

In summary, the $L-$ embedding Hankel and Page matrices over the same time-series signals share the same rank. The optimal hard singular value threshold index n_x also indicated the rank of the $L-$ embedding Hankel matrix constructed over the same time-series signal.

Data-Driven Transition Matrix

The time-delay embedded Hankel matrices are used to compute the data-driven state-transition matrix $A_t^{\hat{x},o} \in \mathbb{R}^{L\times L}$ in (7.11) for lifted observable state $\boldsymbol{h}_t^{\hat{x},L}$. Since both Hankel and Page construction share the same rank, the Page matrix can be employed to recover the rank of the system, and the Hankel matrices for separating the signal from noise. Explicitly, a Hankel matrix can be separated as

$$H_t^{\hat{x},L,T} = \underbrace{\sum_{i=1}^{n_x} \sigma_i^H \boldsymbol{\mu}_i^H \left(\boldsymbol{\nu}_i^H\right)^T}_{\approx\, H_t^{\bar{x},L,T}} + \underbrace{\sum_{\rho=n_x+1}^{L} \sigma_i^H \boldsymbol{\mu}_i^H \left(\boldsymbol{\nu}_i^H\right)^T}_{\approx \Delta H_t^{\hat{x},L,T}}. \tag{7.21}$$

Applying the dehankel function on $H_t^{\bar{x},L,T}$, the denoised x-position measurement can be computed as

$$\begin{aligned} \boldsymbol{h}_{t-1}^{\bar{x},L} &= D_H(H_t^{\bar{x},L,T}, t-L, t-1), \\ \boldsymbol{h}_t^{\bar{x},L} &= D_H(H_t^{\bar{x},L,T}, t-L+1, t), \end{aligned}$$

at time instance t after receiving an obstacle position update. Aiming to recover equation (7.11) with denoise state $\boldsymbol{h}_{t-1}^{\bar{x},L}$ and $\boldsymbol{h}_t^{\bar{x},L}$, the state transition matrix can be extracted using pseudoinverse:

$$A_t^{\bar{x},o} = \boldsymbol{h}_t^{\bar{x},L}(\boldsymbol{h}_{t-1}^{\bar{x},L})^\dagger \in \mathbb{R}^{L\times L}. \tag{7.22}$$

The desired obstacle position update law (7.11)-(7.12) using time-delay embedding can be constructed as the following:

$$\boldsymbol{h}_{t+1}^{\bar{x},L} = A_t^{\bar{x},o}\boldsymbol{h}_t^{\bar{x},L} + \boldsymbol{\omega}_t \tag{7.23}$$

$$x_{t+1}^o = \underbrace{\begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}}_{C} \boldsymbol{h}_{t+1}^{\bar{x},L} + v_{t+1} \tag{7.24}$$

where $\boldsymbol{w}_t \in \mathbb{R}^L$ and $v_{t+1} \in \mathbb{R}$ are the process and observation noises which are both assumed to be zero mean possibly multivariate Gaussian noises with covariance $\mathcal{W}_t$ and $\mathcal{V}_t$. The constant vector $C \in \mathbb{R}^{1\times L}$ projects the lifted states to the position observable to obtain future position predictions. The data-driven linear time-varying update equation (7.23) and (7.24) of an arbitrary obstacle is the proposed solution to Problem 5.

It is important to note that (7.23)-(7.24) still contains process noise and measurement noise despite the lifted states being denoised. Such engineering decision is made to anticipate insufficient denoising, possible poor choices of embedding length L and time window T, discretization uncertainties, and the difference between the consecutive state transition matrix $A_{t+1}^{\bar{x},o}$ and $A_t^{\bar{x},o}$ resulted from extrapolation errors.

Remark 7. *An alternative to* (7.22) *in obtaining* $A_t^{\bar{x},j}$ *using two consecutive denoised Hankel matrix*

$$A_t^{\bar{x},o} = H_t^{\bar{x},L,T}(H_{t-1}^{\bar{x},L,T})^\dagger \in \mathbb{R}^{L\times L} \tag{7.25}$$

which is less sensitive to insufficient signal noise separation and less likely to have ill-conditioned matrix inversion problems. However, operation (7.25) *is numerically expensive and can ignore sudden dynamic changes since it is performing a numerical averaging of $A_t^{\hat{x},o}$ for $t \in \mathbb{Z}_{t-TL:t-1}$.*

Further, the optimal singular value noise and signal separation results can also provide estimates of the process and measurement noise covariance. Apply the dehankel function, the separated measurement noise becomes

$$\boldsymbol{h}_t^{\delta x,TL} = D_H(\Delta H_t^{x,L,T}, t - TL + 1, t).$$

Further, the process and measurement noise covariance matrices can be approximated conservatively based on the empirical variance of the separated measurement noise as

$$\mathcal{W}_t^x \approx \operatorname{diag}((\sigma^{\hat{x}})^2, \ldots, (\sigma^{\hat{x}})^2), \qquad \mathcal{V}_t^x \approx (\sigma^{\hat{x}})^2, \tag{7.26}$$

where $\sigma^{\hat{x},j}$, the unknown white noise level can be estimated using (7.19). The mean of the noise vector is omitted since the measurement noise is assumed to have zero mean. In practice, the mean of the separated measurement noise is not necessarily zero when directly computed the mean of vector $\boldsymbol{h}_{0:TL-1}^{\delta x}$ due to insufficient signal and noise separation or insufficient number of samples to infer the true mean. Nevertheless, by setting the mean to zero, the estimated process and measurement noise covariance is more conservative than the true covariance yielding which is desirable in safety-critical applications.

7.3 Data-Driven Heuristic Unsafe Set

The recent work [201] establishes a connection between the Hankel interpretation of the Koopman operator and with Frenet-Serret frame from differential geometry. In particular, the authors showed that the linear model extracted from time-delay embedding corresponds to the intrinsic curvatures in the Frenet-Serret frame. Following this idea, one can also interpret the matrix $A_t^{\bar{x},j}$ as the Jacobian matrix of the nonlinear system dynamics using a Frenet-Serret coordinate transformation. With such geometry interpretation in mind, a standard nonlinear state estimation technique, the Extended Kalman Filter (EKF), is formulated to estimate the uncertainty associated with the obstacle predictions as new measurements are acquired in streaming data. To clarify notation, indicator $\tilde{\cdot}$ is used for predicted (estimated) values (tilde) and $\hat{\cdot}$ for the raw measured data (hat).

Step 1: Horizon Predictions:

$$\tilde{\boldsymbol{h}}^{\bar{x},L}_{t|t-1} = A^{\bar{x},o}_{t-1}\boldsymbol{h}^{\bar{x},L}_{t-1|t-1} \tag{7.27}$$

is the state prediction where the state covariance matrix is updated as

$$\tilde{\Sigma}^{\bar{x}}_{t|t-1} = A^{\bar{x},o}_{t-1}\Sigma^{\bar{x}}_{t-1|t-1}(A^{\bar{x},o}_{t-1})^T + \mathcal{W}^x_t. \tag{7.28}$$

For all $i \in \{2, \cdots, n_p\}$ where n_p is the predictor horizon, the future states and covariance are predicted as

$$\tilde{\boldsymbol{h}}^{\bar{x},L}_{t+i-1|t} = A^{\bar{x},o}_{t-1}\tilde{\boldsymbol{h}}^{\bar{x},L}_{t+i-2\big|t-1}, \tag{7.29}$$

$$\tilde{\Sigma}^{\bar{x}}_{t+i-1\big|t-1} = A^{\bar{x}}_{t-1}\tilde{\Sigma}^{\bar{x}}_{t+i-2\big|t-1}(A^{\bar{x}}_{t-1})^T + \mathcal{W}^x_t, \tag{7.30}$$

using the most recent state transition matrix available at time $t-1$ and the predicted states at the previous time instance.

Step 2: Next-step measurement update:

$$\tilde{e}^x_t = \hat{x}_t - C\,\tilde{\boldsymbol{h}}^{\bar{x},L}_{t|t-1} \tag{7.31}$$

$$S^x_t = C\,\tilde{\Sigma}^{\bar{x}}_{t|t-1}C^T + \mathcal{V}^x_t \tag{7.32}$$

$$K^x_t = \tilde{\Sigma}^{\bar{x}}_{t|t-1}C^T(S^x_t)^{-1} \tag{7.33}$$

$$\boldsymbol{h}^{\bar{x},L}_{t|t} = \tilde{\boldsymbol{h}}^{\bar{x},L}_{t|t-1} + K^{\bar{x}}_t\tilde{e}^x_t \tag{7.34}$$

$$\Sigma^{\bar{x}}_{t|t} = (I - K^{\bar{x}}_t C)\tilde{\Sigma}^{\bar{x}}_{t|t-1} \tag{7.35}$$

where K^x_t is the optimal Kalman gain that minimizes the measurement residual error $\tilde{e}^x_t$. The term S^x_t is the measurement residual covariance.

The predicted obstacle positions at future time $t+i$ given current time t is a multivariate random variable described by the following first and second moment:

$$\tilde{\boldsymbol{o}}_{t+i} \sim \left(\underbrace{\begin{bmatrix}\mu^x_{t+i|t}\\ \mu^y_{t+i|t}\\ \mu^z_{t+i|t}\end{bmatrix}}_{\triangleq\boldsymbol{\mu}_{t+i|t}}, \underbrace{\begin{bmatrix}\tilde{\Sigma}^x_{t+i|t} & 0 & 0\\ 0 & \tilde{\Sigma}^y_{t+i|t} & 0\\ 0 & 0 & \tilde{\Sigma}^z_{t+i|t}\end{bmatrix}}_{\triangleq\Sigma_{t+i|t}}\right) \tag{7.36}$$

for all $i \in \mathbb{Z}_{1:n_p}$ where $\mu^x_{t+i|t}$, $\mu^y_{t+i|t}$, and $\mu^z_{t+i|t}$ are the predicted obstacle's x, y, z positions which can be extracted as the following $\mu^x_{t+i|t} = C\tilde{\boldsymbol{h}}^{\bar{x},L}_{t+i|t}$. Similarly, $\tilde{\Sigma}^x_{t+i|t}$,

Algorithm 6: Prediction Algorithms (Per Obstacle)

```
Data: Real-time Obstacle center position measurements obtained at time
      instance $t$ under $f_s$ Hz frequency $\hat{x}_t, \hat{y}_t, \hat{z}_t$.
      User-defined constants: $n_p$, $L$.
Result: Forecast predicted obstacle distribution $\{\boldsymbol{\mu}_{t+i|t}\}_{i=1}^{n_p}$, $\{\Sigma_{t+i|t}\}_{i=1}^{n_p}$
while Task is Not Complete do
    for Observables $o = \{x, y, z\}$ do
        Initialized $n_o = -1$.
        while $t \geq 2L$ do
            Construct $L-$embedding lifting vector $\boldsymbol{h}_t^{\hat{o},L}$ and Hankel Matrices
              $H_t^{\hat{o},L,L}$ with the latest $2L$ observables.
            if $t == L^2$ then
                Construct $L-$embedding Page Matrix $P_t^{\hat{o},L,L}$ and use (7.18) to
                  obtain $\tau_o^*$ then $n_o$.
            end
            if $n_o == -1$ then
                Compute $A_t^{\hat{o}}$ using consecutive noisy lifted states $\boldsymbol{h}_t^{\hat{o},L}$ and $\boldsymbol{h}_{t-1}^{\hat{o},L}$
                  following (7.22).
            end
            else
                Perform SVD and signal noise separation of noisy Hankel
                  matrix $H_t^{\hat{o},L,L}$ following (7.21) to obtain matrices $H_t^{\bar{o},L,L}$ and
                  $H_t^{\delta o,L,L}$.
                Apply dehankel operator (7.14) to compute denoised lifted
                  states $\boldsymbol{h}_t^{\bar{o},L}$ with $N_i = t - L + 1$ and $N_f = t$ as well as $\boldsymbol{h}_{t-1}^{\bar{o},L}$ with
                  $N_i = t - L$ and $N_f = t - 1$.
                Compute $A_t^{\bar{o}}$ using (7.22).
                Compute $\mathcal{W}_t$ and $\mathcal{V}_t$ using (7.26)
            end
        end
        Following Steps 1 and 2 to obtain $\mu_{t+i|t}^o$ and $\Sigma_{t+i|t}^o$ for $i \in \mathbb{Z}_1^{n_p}$.
    end
end
```

$\tilde{\Sigma}^y_{t+i|t}$, and $\tilde{\Sigma}^z_{t+i|t}$ are the predicted x, y, z position covariance which can be found by setting $\Sigma^x_{t+i|t} = \tilde{\Sigma}^{\bar{x}}_{t+i|t}$.

It is important to recognize that the assumption that process and measurement noise $\boldsymbol{w}_t$ and v_t are zero-mean Gaussian noises is not necessarily true for all obstacles. The covariance structure in (7.36) also assumes x, y, z measurements are independent. The authors fully acknowledge the limitations of the predicted obstacle statistics and will denote them as “heuristic statistics” to differentiate from later results that

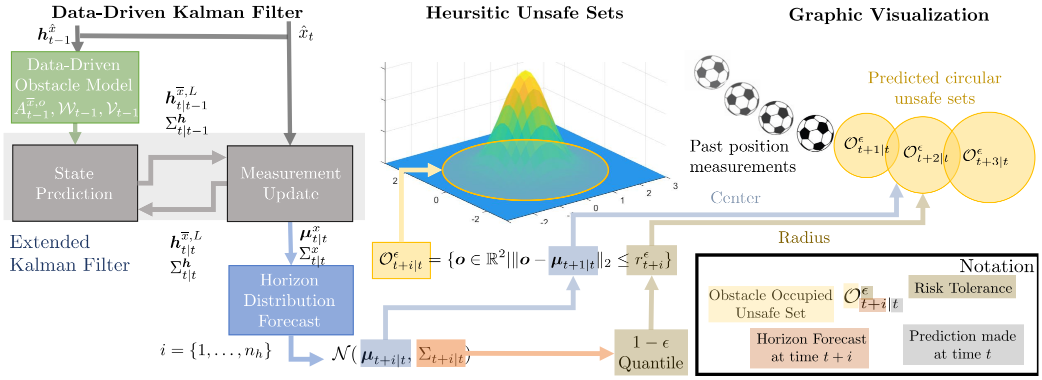
Data-Driven Kalman Filter
$h^{\hat{x}}_{t-1}$
$\hat{x}_t$
Data-Driven Obstacle Model $A^{\bar{x},o}_{t-1}, \mathcal{W}_{t-1}, \mathcal{V}_{t-1}$
$h^{\bar{x},L}_{t|t-1}$
$\Sigma^{h}_{t|t-1}$
State Prediction
Measurement Update
Extended Kalman Filter
$h^{\bar{x},L}_{t|t}$
$\Sigma^{h}_{t|t}$
$\mu^{x}_{t|t}$
$\Sigma^{x}_{t|t}$
Horizon Distribution Forecast
$i = \{1, \ldots, n_h\}$
$\mathcal{N}(\mu_{t+i|t}, \Sigma_{t+i|t})$
Heursitic Unsafe Sets
$\mathcal{O}^{\epsilon}_{t+i|t} = \{o \in \mathbb{R}^2 \,|\, \|o - \mu_{t+1|t}\|_2 \leq r^{\epsilon}_{t+i}\}$
$1-\epsilon$ Quantile
Graphic Visualization
Past position measurements
Predicted circular unsafe sets
$\mathcal{O}^{\epsilon}_{t+1|t}$
$\mathcal{O}^{\epsilon}_{t+2|t}$
$\mathcal{O}^{\epsilon}_{t+3|t}$
Center
Radius
Notation
Obstacle Occupied Unsafe Set
$\mathcal{O}^{\epsilon}_{t+i|t}$
Risk Tolerance
Horizon Forecast at time $t+i$
Prediction made at time t

Figure 7.3: Illustration of the construction of the Heuristic Unsafe Set. Using the data-driven obstacle model, a companion Kalman filter is proposed to make heuristic obstacle prediction estimates. Assume the data-driven time-varying system is perturbed by Gaussian process and measurement noises, horizon distributional forecasts are made as $\mathcal{N}(\boldsymbol{\mu}_{t+i|t}, \Sigma_{t+i|t})$ as described in (7.36). Notation schematics are provided as clarification.

remove these assumptions.

Obstacle Observation Window

The prediction algorithm 6 has two key timings $(2L, TL)$ based on the number of obstacle observations available, referred to as the observation window. For the case $t < 2L$, the observation window is shorter than the activation threshold of the proposed predictor. To provide a backup strategy, the heuristic backup distributions are chosen to be

$$\tilde{\boldsymbol{o}}_{t+i} \sim \left(\begin{bmatrix} \hat{x}_t \\ \hat{y}_t \\ \hat{z}_t \end{bmatrix}, \alpha_{safe} e^{i\Delta t} I_{3\times 3} \right), \quad \forall i \in \mathbb{Z}_{1:n_p}. \tag{7.37}$$

where $\alpha_{safe} \in \mathbb{R}_{>0}$ is a linear safety factor making the unsafe set grow exponentially as the prediction horizon grows, and Δt is the planning time step. The back heuristic unsafe sets (7.37) mimic the standard potential field approach with the dynamic obstacles approximated to be stationary during the prediction horizon. By compounding the uncertain radius, correlated to the covariance, the unsafe set expands as longer predictions are made to promote safer obstacle-avoiding behavior until more observations are acquired over time.

For the time window between $2L - 1$ to TL, the proposed algorithm has acquired enough data to deduce the linear propagator $A^{\bar{x},o}_t$ for making position predictions. However, the limited observations are insufficient to construct the page matrix for optimal denoising and rank reduction. Given the data-driven transition matrix can be less accurate and lacks initialization to the process noise covariance, $\mathcal{W}_t$, and measurement noise covariance, $\mathcal{V}_t$, a conservative initial process and measurement noise covariance is recommended to formulate a more conservative heuristic unsafe set. Once the number of observables exceeds the desired threshold TL, the proposed algorithm will be able to perform optimal denoising as well as contracting the process and measurement noise covariance to avoid being overly conservative.

Heuristic Unsafe Set

Given predicted obstacle position probability distributions (7.36) or (7.37), a deterministic unsafe set can be constructed given risk-tolerance ϵ using the quantile function of multivariate normal distributions. Let $\mathbb{Q}_{\mathcal{N},1-\epsilon}$ denote the $(1-\epsilon)$ Quantile Z-score of a normal Gaussian distribution with zero mean and unit standard deviation. the obstacle heuristic unsafe set becomes

$$\mathcal{O}^{\epsilon}_{t+i|t} = \{\boldsymbol{o} \in \mathbb{R}^3 | \|\boldsymbol{o} - \boldsymbol{\mu}_{t+i|t}\|_2 \le r^{\epsilon}_{t+i}\} \tag{7.38}$$

where the heuristic unsafe set radius given $\epsilon \in (0, 1)$ is

$$r^{\epsilon}_{t+i} = \mathbb{Q}_{\mathcal{N},1-\epsilon} \max\{\sqrt{\Sigma^x_{t+i|t}}, \sqrt{\Sigma^y_{t+i|t}}, \sqrt{\Sigma^z_{t+i|t}}\} \tag{7.39}$$

for each $i \in \mathbb{Z}_{1:n_p}$ as illustrated in Fig. 7.3.

7.4 Conformal Unsafe Set

In Chapter 6, adaptive conformal prediction is applied to quantify the uncertainty of arbitrary horizon predictor. In this work, a similar idea is borrowed to provide one-step prediction uncertainty quantification of the uncalibrated predictor proposed in Section 7.2, given dependent, time series obstacle position measurements.

The choice of nonconformity score is crucial which needs a metric that reflects predictor accuracy and the resulting conformity prediction set given user-defined risk tolerance can be translated into obstacle-avoiding statements. Straightforwardly, the Euclidean distance between the predicted obstacle position relative to the measurements acquired can be used as the nonconformity score, mathematically

$$\beta_{t+1|t} \triangleq \|\boldsymbol{\mu}_{t+1|t} - \hat{\boldsymbol{o}}_{t+1}\|_2. \tag{7.40}$$

Similarly, multi-step prediction nonconformity score $\beta_{t+i|t} \triangleq \|\boldsymbol{\mu}_{t+i|t} - \hat{\boldsymbol{o}}_{t+i}\|_2$ that captures the prediction error at time t made for future time steps $i \in \mathbb{Z}_1^{n_p}$. Intuitively, a large nonconformity score indicates that the position prediction $\boldsymbol{\mu}_{t+i|t}$ is inaccurate based on its measurement feedback, while a small score indicates an accurate prediction.

However, it is important to note that there are no obstacle position measurements available for future measurements for time $t+i$ given current time t. To address this issue, recall *time-lagged nonconformity score* from Chapter 6:

$$\beta_{t|t-i} = \|\boldsymbol{\mu}_{t|t-i} - \hat{\boldsymbol{o}}_t\|_2, \qquad \forall i \in \mathbb{Z}_1^{n_p} \tag{7.41}$$

which can be evaluated at time t and applied to the update rule (2.12). This nonconformity scores $\beta_{t|t-i}$ are time-lagged in the sense that, at time t, the i step-ahead prediction error is calculated that was made i time steps ago.

Naturally, in the time series prediction setting, nonconformity scores $\beta_{t+1|t-i}$ and $\beta_{t+1|t-j}$ for $i \neq j$ are not independent and identically distributed. However, by adapting the prediction regions based on the performance of the multi-step predictions using adaptive conformal prediction, on-average coverage guarantees can be made, albeit weaker than the point-wise guarantees provided by conformal predictions.

For each time step t, a prediction-driven unsafe radius ρ_{t+1}^{ϵ} is computed that is associated by an update variable ϵ_{t+1} using update law (2.12). Using algorithm 7, adaptive from [177], a spherical conformal unsafe set can be constructed for all $t > N_{acp}$ for every obstacle.

Theorem 8. *[Adapted from [177], Theorem 3] Let γ be a learning rate, $\epsilon_0 \in (0,1)$ be an initial value for the recursion*

$$\epsilon_{t+1} \triangleq \epsilon_t + \gamma(\epsilon - \mathbf{1}_{\|\boldsymbol{\mu}_{t|t-1} - \hat{\boldsymbol{o}}_t\|_2 > \rho_t^{\epsilon}}), \tag{7.42}$$

where $\epsilon \in (0,1)$ is the user-specific risk tolerance. Let T be the number of times that the recursion (7.42) *is computed. For the one-step-ahead prediction errors, it holds that*

$$b_1 \leq \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{P}[\|\boldsymbol{\mu}_{t+1|t} - \hat{\boldsymbol{o}}_{t+1}\|_2 \leq \rho_{t+1}^{\epsilon}] \leq b_2 \tag{7.43}$$

where $b_1 \triangleq 1-\epsilon-\frac{\epsilon_0+\gamma}{T\gamma}$, $b_2 \triangleq 1-\epsilon+\frac{(1-\epsilon_0)+\gamma}{T\gamma}$ so that $\lim_{T\to\infty} b_1 = \lim_{T\to\infty} b_2 = 1-\epsilon$.

Proof. See Chapter 6. □

However, a greedy learning rate ($\gamma > 1$) is required for guaranteeing coverage but a large learning rate can lead to the rapid expansion of the updated law (2.12) as observed in the simulation results in Chapter 6. In the robotic application with torque constraints, the sudden growth of a state constraint (obstacle avoidance) can lead to numerical infeasibilities. As a result, only single-step (immediate) prediction region $\rho^{\epsilon}_{t+1|t}$ is considered for the next step forecasts using the horizon predictor. To preserve safe obstacle avoidance guarantees, the conformal prediction of the unsafe set is considered as the safety constraint in synthesizing the obstacle avoidance controls strategy for $t+1$. For the remaining planning horizon, the heuristic unsafe set is accounted for.

Algorithm 7: Conformal Unsafe Set (per Obstacle)

```
Data: User-defined failure probability $\epsilon$, learning rates $\gamma$, Obstacle's one-step
      position prediction $\boldsymbol{\mu}_{t-j|t-j-1}$ from past predictors at time instance
      $t-j-1$ where $j \in \mathbb{Z}_0^{N_\tau}$, Obstacle current position measurement $\hat{\boldsymbol{o}}_t$.
Result: Conformal Unsafe set $\mathcal{O}^{\epsilon,c}_{t+1}$
while Task is not Complete do
    Obtain latest obstacle measurements $\hat{\boldsymbol{o}}_t$. Using horizon predictor
      Algorithm 6 to predict obstacle position at $t+1$ noted as $\boldsymbol{\mu}_{t+1|t}$.
    for j from 1 to $N_\tau$ do
        # Compute ACP regions Use recursion (7.42) to compute $\epsilon_t$,
        Update the nonconformity scores, $\beta_{t-j|t-j-1} = \|\boldsymbol{\mu}_{t-i|t-i-1} - \hat{\boldsymbol{o}}_{t-i}\|_2$.
        Compute index $q_{t-j} \longleftarrow \lceil (N_\tau + 1)(1-\epsilon_t) \rceil$
    end
    The one-step predicted unsafe set radius is $\rho^{\epsilon}_{t+1}$ as the $q^{th}$ smallest value of
      list $\{\beta_{t|t-1}, \ldots, \beta_{t-N_\tau|t-N_\tau-1}\}$.
    The conformal unsafe set can be constructed using (7.44).
end
```

Using the predicted unsafe set radius at time $t+1$ from Algorithm 7, the $1-\epsilon$ confidence conformal time-varying unsafe set becomes

$$\mathcal{O}^{\epsilon,c}_{t+1|t} = \{\boldsymbol{o} \in \mathbb{R}^3 | \|\boldsymbol{o} - \boldsymbol{\mu}_{t+1|t}\|_2 \le \rho^{\epsilon}_{t+1}\}. \tag{7.44}$$

Note, the conformal unsafe set $\mathcal{O}^{\epsilon,c}_{t+1|t}$ for time $t+1$ and the heuristic unsafe sets $\{\mathcal{O}^{\epsilon}_{t+i|t}\}_{i=2}^{n_p}$ made at time t forms the prediction-driven unsafe set. Most importantly, let $\epsilon \in (0,1)$ be a risk tolerance, the conformal unsafe set exhibits the following relation relative to the true unsafe set:

$$\forall \boldsymbol{o} \in \mathcal{O}^{\epsilon,c}_{t+1|t}, \quad \mathbb{P}\left[\boldsymbol{o} \in \mathcal{O}_{t+1}\right] \ge 1-\epsilon, \tag{7.45}$$

which is equivalent to the desired prediction unsafe set inequality (7.8). Therefore, the conformal unsafe set (7.44) is a solution to Problem 6.

Based on the choice of the risk tolerance $\epsilon \in (0,1)$, the case of $\epsilon \longrightarrow 0$ means to avoid the obstacle robustly at the cost of conservatism. The $\epsilon \longrightarrow 1$ case is equivalent to only considering avoiding the predicted trajectory itself. However, the predicted trajectory accuracy cannot be rigorously proved because of learned model uncertainty due to insufficient data and lack of ground truth. Therefore, set predictions are made which equips with statistical significance to provide probabilistic safety guarantees in planning and control in later sections.

7.5 Prediction-Based MPC

Using the proposed data-driven horizon predictor (Algorithm 6), spherical heuristic unsafe sets at future time instance $\{t+i\}_{i=1}^{n_h}$ can be acquired given current time t for each obstacle. In addition, following Algorithm 7, a one-step conformal unsafe set can be acquired for each obstacle that adapts in an online manner while removing the heuristic assumptions made for formulating the heuristic unsafe sets. In this section, the main planning algorithm is presented and analyzed which plans safe trajectories for a prescribed risk-level ϵ, the solution to Problems 7 and 8. Based upon ideas from finite-time optimal control, the proposed optimization-based planning algorithm is efficiently reformulated into a quadratic program. In the presence of obstacles, the safe set at time t, denoted as $\mathcal{S}_t$, being the complement set of the obstacle occupied set, is defined using the ego vehicle current state $\boldsymbol{x}_t$ as

$$\mathcal{S}_t = \{\boldsymbol{x}_t \in \mathcal{D}^{\boldsymbol{x}} | M\boldsymbol{x}_t \notin \cup_{j=1}^{n_{obs}} O_t^j\}. \tag{7.46}$$

where O_t^j is the true occupied unsafe set by obstacle j at time t that is not unknown. Using the proposed data-driven algorithms, a deterministic obstacle unsafe set prediction given a risk-tolerance ϵ is made at time t for the next time instance $O_{t+1|t}^{\epsilon,c}$ using conformal prediction (Algorithm 7) for each obstacle. Since each obstacle's conformal predicted unsafe sets $O_{t+1}^{\epsilon,c}$ are spherical, its complement set is concave in terms of ego vehicle positions. To leverage the efficiency of convex optimization, the following lemma is introduced to reformulate concave probabilistic obstacle avoidance constraints into an affine state constraint.

Lemma 5. *At current time instance t, and under risk-tolerance ϵ, let the spherical set O_{t+1} be the true unsafe set for an obstacle at time i for all $i \in \mathbb{Z}_{t_0}^t$ where $t - t_0 \to \infty$. The following set is*

$$\mathcal{S}_{t+1}^{\epsilon} \triangleq \{\boldsymbol{x}_{i+1} \in \mathcal{D}^{\boldsymbol{x}} \Big| \frac{1}{t-t_0} \left(\sum_{i=t_0+1}^{t} \mathbb{P}[M\boldsymbol{x}_{i+1} \in O_{i+1}] \right) \leq \epsilon\} \tag{7.47}$$

an $(1-\epsilon)$ percent confidence average obstacle-free safe set[3]*. Let $\overline{\boldsymbol{x}}_{i+1} \in \mathcal{D}^{\boldsymbol{x}}$ be the*

[3] If the ego vehicle state $\boldsymbol{x}_i$ where $i \in \mathbb{Z}_{t_0+1}^t$ satisfying the above set, then the ego vehicle is $1-\epsilon$ probability obstacle avoidance on average by definition.

nominal state at time $i+1$*. Then, if for* $i \in \mathbb{Z}_{t_0+1}^{t}$ *the ego vehicle state* $\boldsymbol{x}_{i+1}$ *satisfying the following affine inequality constraint,*

$$\frac{(M\overline{\boldsymbol{x}}_{i+1} - \boldsymbol{\mu}_{i+1|i})^T(M\boldsymbol{x}_{i+1} - \boldsymbol{\mu}_{i+1|i})}{\|M\overline{\boldsymbol{x}}_{i+1} - \boldsymbol{\mu}_{i+1|i}\|_2} \geq \rho_{i+1}^{\epsilon}, \tag{7.48}$$

where $\boldsymbol{\mu}_{i+1|i}$ *is the one-step obstacle prediction using horizon prediction using Algorithm 6 and* ρ_{i+1}^{ϵ} *is obtained by Algorithm 7, is sufficient for the ego vehicle to guarantee* $1-\epsilon$ *probability obstacle avoidance.*

Proof. Leveraging the result of theorem (8) with $t - t_0 \longrightarrow \infty$ where $b_1 = b_2 = 1 - \epsilon$, the following statement is true

$$\frac{1}{t-t_0}\sum_{i=t_0+1}^{t}\mathbb{P}\big[\|\hat{\boldsymbol{o}}_{i+1} - \boldsymbol{\mu}_{i+1|i}\|_2 \leq \rho_{i+1}^{\epsilon}\big] = 1-\epsilon,$$

where ρ_{i+1}^{ϵ} is the conformal unsafe set radius for the one-step prediction made at time instance i using algorithm 7. Then the complement to the conformal unsafe set $O_{i+1}^{\epsilon,c}$ in $\mathbb{R}^3$, is equivalent to the desired safe set (7.47) using properties (7.45).

Geometrically speaking, the largest convex complement to a spherical set is a half-space tangential to the spherical set. Exploiting this idea, the half-space constraint is constructed, *i.e.* the convex safe set, by making a conservative approximation of the separating tangent plane using past previous iteration results. Similar to the sequential convexification reviewed in 2, the parameter $\overline{\theta}_{i+1}$ is defined as the angle between the vectors $\boldsymbol{v}_{i+1} \triangleq M\boldsymbol{x}_{i+1} - \boldsymbol{\mu}_{i+1|i}$ and $\overline{\boldsymbol{v}}_{i+1} \triangleq M\overline{\boldsymbol{x}}_{i+1} - \boldsymbol{\mu}_{i+1|i}$. Leveraging the definition of the dot product, the following is true

$$\|\boldsymbol{v}_{i+1}\|\cos(\overline{\theta}_{i+1}) = \frac{\overline{\boldsymbol{v}}_{i+1}^T\boldsymbol{v}_{i+1}}{\|\overline{\boldsymbol{v}}_{i+1}\|_2}. \tag{7.49}$$

Since $\cos(\overline{\theta}_{i+1}) \leq 1$, by setting $\|\boldsymbol{v}_{i+1}\|\cos(\overline{\theta}_{i+1}) \geq \rho_{i+1}^{\epsilon}$, then all $\boldsymbol{x}_{i+1} \in \mathcal{D}^{\boldsymbol{x}}$ the following inequality relationship holds

$$\|M\boldsymbol{x}_{i+1} - \boldsymbol{\mu}_{i+1|i}\| \geq \rho_{i+1}^{\epsilon}.$$

Therefore, enforcing affine constraint

$$(M\overline{\boldsymbol{x}}_{i+1} - \boldsymbol{\mu}_{i+1|i})^T(M\boldsymbol{x}_{i+1} - \boldsymbol{\mu}_{i+1|i}) \geq \rho_{i+1}^{\epsilon}\|M\overline{\boldsymbol{x}}_{i+1} - \boldsymbol{\mu}_{i+1|i}\|_2,$$

$\forall i \in \mathbb{Z}_{t_0+1}^{t}$ where $t - t_0 \longrightarrow \infty$ is a sufficient condition to set constraint (7.47). □

The nominal state $\overline{\boldsymbol{x}}_{i+1}$ can be obtained from previous solutions given at time $i-1$ as the solution. Explicitly, the nominal state satisfies $\overline{\boldsymbol{x}}_{i+1} = \boldsymbol{x}_{i+1|i}$ following standard warm start rules, see Chapter 2 for more details.

Remark 8. *In Lemma 5, the convex (affine) safe set given by inequality constraint* (7.48) *is the largest possible convex safe set for all* $\boldsymbol{x}_{i+1} \in \mathcal{D}^{\boldsymbol{x}}$ *if* $\|\boldsymbol{x}_{i+1} - \overline{\boldsymbol{x}}_{i+1}\| \to 0$ *since* $\cos \overline{\theta}_{i+1} \to 1$.

For real-time obstacle avoidance, the averaging sample $t - t_0 \to \infty$, in lemma 5, implies the need for infinite time obstacle observations and averaging which is unpractical. Therefore, the following Lemma is constructed, an extension of lemma (5) to finite $t - t_0 \in \mathbb{Z}_{>0}$.

Lemma 6. *At current time instance t, given a user-defined risk-level* $\epsilon \in (0, 1]$, *a learning rate* γ, *let* $\overline{\boldsymbol{x}}_{i+1} \in \mathcal{D}^{\boldsymbol{x}}$ *be the nominal state at time* $i+1$ *for all* $i \in \mathbb{Z}^t_{t_0+1}$. *Let* $\epsilon_0 \in (0, 1)$ *be an initial value for the recursion* (7.42)*, and a finite* $t - t_0$ *be the number of times that recursion* (7.42) *is computed where* $\lceil \frac{\epsilon_0+\gamma}{\gamma\epsilon} \rceil < t - t_0 \leq t + 1$. *Consider an augmented risk-level*

$$\overline{\epsilon} = \epsilon - \frac{\epsilon_0 + \gamma}{(t - t_0)\gamma} \tag{7.50}$$

where $\overline{\epsilon} \leq \epsilon$. *Let* $t > t_0$, *define the probabilistic obstacle avoidance safe set on average as*

$$\mathcal{S}^f_{t+1} \triangleq \{\boldsymbol{x}_{i+1} \in \mathcal{D}^{\boldsymbol{x}} \left| \frac{1}{t - t_0} \left(\sum_{i=t_0+1}^{t} \mathbb{P}[M\boldsymbol{x}_{i+1} \in \mathcal{O}_{i+1}] \right) \leq \epsilon \right\}. \tag{7.51}$$

If the ego vehicle state $\boldsymbol{x}_{i+1}$ *where* $i \in \mathbb{Z}^t_{t_0+1}$ *satisfying the above set for finite* $t - t_0$, *then the ego vehicle is* $1 - \epsilon$ *probability obstacle avoidance on average for finite time by definition.*

Now, using the augmented risk-level $\overline{\epsilon}$ *in recursion* (7.42) *instead of* ϵ, *an augmented the conformal unsafe set radius* $\overline{\rho}^{\overline{\epsilon}}_{i+1}$ *is computed. The feasible set of the following affine inequality constraint is as a subset of* $\mathcal{S}^f_{t+1}$,

$$\frac{(M\overline{\boldsymbol{x}}_{i+1} - \boldsymbol{\mu}_{i+1|i})^T (M\boldsymbol{x}_{i+1} - \boldsymbol{\mu}_{i+1|i})}{\|M\overline{\boldsymbol{x}}_{i+1} - \boldsymbol{\mu}_{i+1|i}\|_2} \geq \overline{\rho}^{\overline{\epsilon}}_{i+1}. \tag{7.52}$$

$\forall i \in \{t_0 + 1, t\}$.

Proof. Leveraging the result of (7.43) and empirical quantile function is non-decreasing, the following inequality holds

$$\Big\{\boldsymbol{x}_{i+1} \in \mathcal{D}^{\boldsymbol{x}} \Big| \sum_{i=t_0+1}^{t} \mathbb{P}\big[\, \|M\boldsymbol{x}_{i+1} - \boldsymbol{\mu}_{i+1|i}\|_2 \leq (t-t_0)\rho_{i+1}^{\overline{\epsilon}}\, \big]\Big\}^c \subseteq$$

$$\Big\{\boldsymbol{x}_{i+1} \in \mathcal{D}^{\boldsymbol{x}} \Big| \sum_{i=t_0+1}^{t} \mathbb{P}[\|M\boldsymbol{x}_{i+1} - \boldsymbol{\mu}_{i+1|i}\|_2 \leq (t-t_0)\rho_{i+1}^{\epsilon}]\Big\}^c \subseteq \mathcal{S}_{t+1}^{f}$$

because $\rho_{i+1}^{\overline{\epsilon}} \geq \rho_{i+1}^{\epsilon}$ for each $i \in \mathbb{Z}_{t_0+1}^{t}$ given $\overline{\epsilon} \leq \epsilon$ where the superscript $\{\cdot\}^c$ stands for set complement. Therefore, by making a conservative convexification over inequality constraint $\|M\boldsymbol{x}_{i+1} - \boldsymbol{\mu}_{i+1|i}\|_2 \geq \rho_{i+1}^{\epsilon}$ suffices the safety requirement. The remaining proof for obtaining the affine constraint follows the proof of Lemma 5. □

Remark 9. *There is one implicit requirement in Lemma 6 that is worth remarking. The augmented risk-level* (7.50) *must be within* $(0, 1)$ *where* $\epsilon > \frac{\epsilon_0+\gamma}{(t-t_0)\gamma}$. *Based on the choice of initial risk level* ϵ_0*, learning rate* γ*, and desired risk level* ϵ*, Lemma 6 requires* $t - t_0 > \lceil\frac{\epsilon_0+\gamma}{\gamma\epsilon}\rceil$.

Alternative to using the adaptive conformal prediction to acquire probabilistic obstacle avoidance on average, the extended Kalman filter given in the horizon predictor also provides a heuristic unsafe set $\mathcal{O}_{t+i|t-1}^{\epsilon}$ for $i \in \mathbb{Z}_0^{n_h-1}$ given a risk tolerance $\epsilon \in (0, 1)$. Under the assumption that the process and measurement noise in the lifted model (7.23)-(7.24) are Gaussian random variables. Heuristic affine inequality constraints can be similarly constructed as

$$\frac{(M\overline{\boldsymbol{x}}_{t+i} - \boldsymbol{\mu}_{t+i|t})^T(M\boldsymbol{x}_{t+i} - \boldsymbol{\mu}_{t+i|t})}{\|M\overline{\boldsymbol{x}}_{t+i} - \boldsymbol{\mu}_{t+i|t}\|_2} \geq r_{t+i}^{\epsilon} \tag{7.53}$$

where r_{t+i}^{ϵ} is defined in (7.39) for each $i \in \mathbb{Z}_1^{n_h}$ at current time t. The assumption that the joint normal distributions defined by $\mathcal{N}_t \times \mathcal{N}_{t+1} \times \cdots \times \mathcal{N}_{t+n_h}$ is largely not true without stringent assumptions on obstacle behaviors, data availability, and dependency assumptions. Further, the spherical obstacle unsafe set expands exponentially as the prediction horizon increases. For practical implementation, the heuristic obstacle avoidance constraints (7.53) for $t + 2$ to $t + n_h$ are considered with slacking to reduce infeasibility.

In Summary, the affine inequalities constraints (7.48), (7.52), and (7.53) are the proposed data-driven solution to Problem 7.

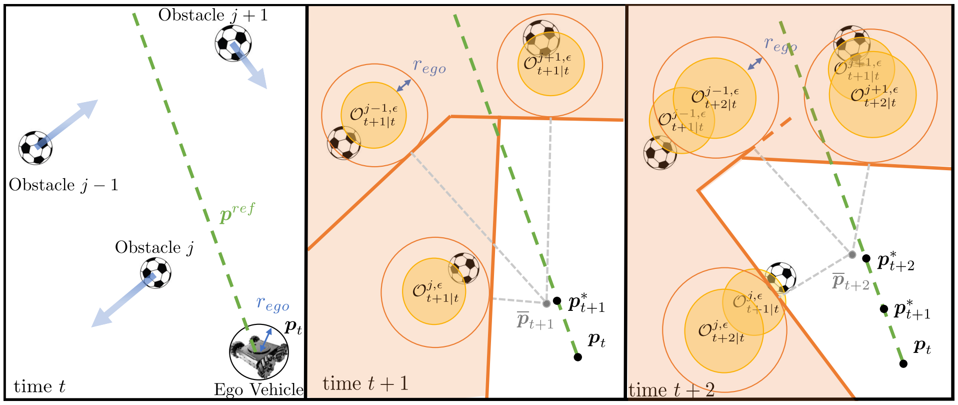
Obstacle $j+1$
Obstacle $j-1$
p^{ref}
Obstacle j
r_{ego}
p_t
time t
Ego Vehicle
r_{ego}
$\mathcal{O}^{j-1,\epsilon}_{t+1|t}$
$\mathcal{O}^{j+1,\epsilon}_{t+1|t}$
$\mathcal{O}^{j,\epsilon}_{t+1|t}$
p^*_{t+1}
$\overline{p}_{t+1}$
p_t
time $t+1$
r_{ego}
$\mathcal{O}^{j-1,\epsilon}_{t+2|t}$
$\mathcal{O}^{j-1,\epsilon}_{t+1|t}$
$\mathcal{O}^{j+1,\epsilon}_{t+1|t}$
$\mathcal{O}^{j+1,\epsilon}_{t+2|t}$
p^*_{t+2}
$\overline{p}_{t+2}$
p^*_{t+1}
$\mathcal{O}^{j,\epsilon}_{t+1|t}$
$\mathcal{O}^{j,\epsilon}_{t+2|t}$
p_t
time $t+2$

Figure 7.4: Illustration of the sequential convexification using hyperplane of the predicted unsafe sets. Set spherical set $O^{\epsilon}_{t+i|t}$ is the $t+i$ time forecast of the obstacle occupied unsafe set given a risk tolerance of $\epsilon \in (0,1)$ where the forecast is made at time t. r_{ego} is the radius of the smallest spherical convex hall encapsulating the ego vehicle which is added to the predicted unsafe sets. The hyperplane is selected to be tangential to the spherical unsafe sets defined by a normal vector pointing from the center of the spherical unsafe set to the previously computed ego vehicle position $\overline{\boldsymbol{p}}_{t+i}$.

Single Obstacle Formulation

Given the data-driven unsafe sets and their convex reformulations, the proposed solution to Problem 7 is a prediction-aware model predictive controller (PA-MPC). It is synthesized by solving a finite-time optimal control problem constrained by the data-driven predicted safe sets. Thanks to the affine safety constraints, the linear, discrete time-varying system governed by (7.1) - (7.2) can guarantee probabilistic obstacle avoidance on average in finite-time by issuing the solution of the following sequential Quadratic Program (QP). The following optimization problem is solved at every time instance t for a optimal input sequence $\{\boldsymbol{u}^*_{t+i}\}_{i=0}^{n_h-1}$ and slack variables $\{\boldsymbol{\delta}^*_{t+i}\}_{i=0}^{n_h-1}$. Such optimal input sequence $\{\boldsymbol{u}^*_{t+i}\}_{i=0}^{n_h-1}$ are the solution to the probabilistic safe motion planning problem (8). To incorporate state feedback, only the first input $\boldsymbol{u}_t$ will be rolled out at time instance t, and the following QP (7.55) will be solved again with updated initial conditions and predicted sets.

Theorem 9. *Consider Problem 7 with system state and output dynamics* (7.1) – (7.2) *where a single moving obstacle is present with unknown dynamics, intention. The obstacle position measurement is observable in real-time satisfying the described behaviors in Problem 5. Suppose the ego vehicle is tasked to track a reference position trajectory* $\{\boldsymbol{p}^{ref}_t\}_{t=0}^{\infty}$ *where* $\boldsymbol{p}^{ref}_t \in \mathbb{R}^3$ *that is designed regardless of the moving obstacle. Given convex state and input admissible sets* $\mathcal{D}^{\boldsymbol{x}}$ *and* $\mathcal{D}^{\boldsymbol{u}}$*, respectively. A* *convex stage cost function* $\mathcal{L}_i(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{p}^{ref}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^3 \longrightarrow \mathbb{R}$ *of the form:*

$$\mathcal{L}_i = \frac{1}{2}\left((M\boldsymbol{x}_i - \boldsymbol{p}_i^{ref})^T Q (M\boldsymbol{x}_i - \boldsymbol{p}_i^{ref}) + \boldsymbol{u}_i^T R \boldsymbol{u}_i\right) \tag{7.54}$$

where matrices $Q \in \mathbb{R}^{3\times3}$ *and* $R \in \mathbb{R}^{n_u \times n_u}$ *are positive definite.*

A recursively feasible solution to the following sequential quadratic program provides probabilistic obstacle avoidance on average in finite time. The quadratic program at iteration ω^{th} *is the following:*

$$\min_{\substack{\{\boldsymbol{u}_i\}_{i=0}^{n_h-1} \\ \{\delta_i\}_{i=0}^{n_h-1}}} \sum_{i=0}^{n_h-1} \mathcal{L}_i(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{p}^{ref}) + \zeta_{0,i}\delta_i^2 + \zeta_{1,i}\delta_i \tag{7.55a}$$

$$s.t. \quad \boldsymbol{x}_{i+1} = A_{t+i}\boldsymbol{x}_i + B_{t+i}\boldsymbol{u}_i + C_{t+i} \tag{7.55b}$$

$$\boldsymbol{\alpha}_{t+i|t}\boldsymbol{x}_{i+1} + \beta_{t+i|t} \leq \delta_i \tag{7.55c}$$

$$\boldsymbol{x}_i \in \mathcal{D}^{\boldsymbol{x}} \tag{7.55d}$$

$$\boldsymbol{u}_i \in \mathcal{D}^{\boldsymbol{u}} \tag{7.55e}$$

$$\mathbf{1}^T(\boldsymbol{x}_{i+1} - \overline{\boldsymbol{x}}_{i+1}) \leq m_x \tau_x^w \tag{7.55f}$$

$$\mathbf{1}^T(\overline{\boldsymbol{x}}_{i+1} - \boldsymbol{x}_{i+1}) \leq m_x \tau_x^w \tag{7.55g}$$

$$\mathbf{1}^T(\boldsymbol{u}_i - \overline{\boldsymbol{u}}_i) \leq m_u \tau_u^w \tag{7.55h}$$

$$\mathbf{1}^T(\overline{\boldsymbol{u}}_i - \boldsymbol{u}_i) \leq m_u \tau_u^w \quad \forall k \in \mathbb{Z}_0^{n^h-1} \tag{7.55i}$$

$$\boldsymbol{x}_0 = \boldsymbol{x}_t \tag{7.55j}$$

$$\delta_0 = 0, \quad \{\delta_i\}_{i=1}^{n_h-1} \geq 0 \tag{7.55k}$$

where $\{\overline{\boldsymbol{x}}_k\}_{k=1}^{n_h}$*, and* $\{\overline{\boldsymbol{u}}_k\}_{k=0}^{n_h-1}$ *are the* $(w-1)^{th}$ *iteration solution of this Quadratic Program. Parameters* m_x*,* m_u*,* $\{\zeta_{0,i}\}_{i=0}^{n_h-1}$ *and* $\{\zeta_{1,i}\}_{i=0}^{n_h-1}$ *are adjustable positive constants. Parameters* $\tau_x, \tau_u \in (0,1)$[4]*. Coefficients* $\boldsymbol{\alpha}_{t+i|t} \in \mathbb{R}^{1\times n_x}$ *in constraint* (7.55c) *can be computed using the predicted obstacle trajectory using the proposed horizon predictor*

$$\boldsymbol{\alpha}_{t+i|t}^T = -M^T \underbrace{\frac{M\overline{\boldsymbol{x}}_{t+i+1} - \boldsymbol{\mu}_{t+i+1|t}}{\|M\overline{\boldsymbol{x}}_{t+i+1} - \boldsymbol{\mu}_{t+i+1|t}\|_2}}_{\triangleq \boldsymbol{\lambda}_{t+i+1|t}},$$

where $i \in \mathbb{Z}_{i=0}^{n_h-1}$*. For* $i = 0$*, parameter* $\beta_{t|t} \in \mathbb{R}$ *in constraint* (7.55c) *is derived using the one-step conformal unsafe set radius* $\rho_{t+1}^{\bar{\epsilon}}$ *given current time* t*,*

$$\beta_{t+1|t} = \boldsymbol{\lambda}_{t+1|t}^T M \boldsymbol{\mu}_{t+1} + \rho_{t+1}^{\bar{\epsilon}}. \tag{7.56}$$

[4] At $t = 0$, the 0^{th} iteration $\{\overline{\boldsymbol{x}}_k\}_{k=1}^{n_h}$ and $\{\overline{\boldsymbol{u}}_k\}_{k=0}^{n_h-1}$ is known as the warm start which can be found offline.

For $i \in \mathbb{Z}^{1:n_h-1}$, parameters $\beta_{t+i|t} \in \mathbb{R}$ are obtained from the heuristic unsafe set (7.53),

$$\beta_{t+i|t} = \boldsymbol{\lambda}^T_{t+i+1|t} M \boldsymbol{\mu}_{t+i+1|t} + r^{\epsilon}_{t+i+1}. \tag{7.57}$$

Lastly, the coefficients $\{\lambda_{0,i}\}_{i=0}^{n_h-1}$ and $\{\lambda_{1,i}\}_{i=0}^{n_h-1}$ are positive definite constants.

Proof. Since only the first input of the sequence $\{\boldsymbol{u}^*_{t+i}\}_{i=0}^{n_h}$ is being implemented at time instance t in a receeding horizon fashion. This proof only assesses the vehicle's safety condition at time $t+1$ given initial state $\boldsymbol{x}_t$ and control input $\boldsymbol{u}^*_t$.

From the result of Lemma 6, the satisfying inequality (7.52) at time $t+1$ guarantees the probability of the ego vehicle at time $t+1$ entering in the unsafe set occupied by the obstacle on finite-time average over is less than ϵ. Therefore, a feasible input $\boldsymbol{u}_t$ satisfying constraint (7.55c) given initial condition $\boldsymbol{x}_t$ guarantees probabilistic obstacle avoidance on average in finite time.

Combining affine inequality constraints (7.55f) - (7.55h) is equivalent to the 1-norm constraint $\|\boldsymbol{x}_{t+i+1} - \overline{\boldsymbol{x}}_{t+i+1}\|_1 \longrightarrow 0$ and $\|\boldsymbol{u}_{t+i} - \overline{\boldsymbol{u}}_{t+i}\|_1 \longrightarrow 0$. By picking an initial trust region m_x and m_u and decay rate $\tau_x > 1$ and $\tau_u > 1$, as $\omega \longrightarrow \infty$ the inequality constraints (7.55f) – (7.55h) forces the sequential convexification of the obstacle avoidance constraint to converge. [5] Further, by the proof of Lemma 5 and summarized in Remark 8, as the iteration counter $\omega \longrightarrow \infty$ achieves the tightest convex reformulation. □

The positive semi-definite slack variables $\{\delta_i\}_{i=0}^{n_h-1}$ for the obstacle avoidance constraint (7.55c) at future time instances $t+1$ to $t+n_h-1$ are introduced to balance the safety informed by the heuristic unsafe set and tracking performances. To avoid overly conservative trajectories as well as lack of feasible inputs balancing trust regions constraints (7.55f) – (7.55i), state and torque bounds (7.55d) – (7.55e), and obstacle avoidance constraints (7.55c), the authors strongly recommend an additive slack variables δ_i for future instances $t+i$ to $t+n_h-1$. The associated slack variable costs $\{\zeta_{0,i}\}_{i=1}^{n_h-1}$ and $\{\zeta_{1,i}\}_{i=1}^{n_h-1}$ can be chosen to encourage or discourage the program to avoid the heuristic unsafe sets over reference tracking.

[5] Therefore, for real-time robotic applications, the users can terminate the sequential quadratic program (7.55) in finite iterations.

Multiple Obstacles Formulation

In the presence of multiple moving obstacles, additional obstacle avoidance constraints like (7.55c) can be appended into the PA-MPC optimization for each additional obstacle. At the expanse of additional affine state constraints and auxiliary optimization variables for slacking, the QP (7.55c) can be modified to accommodate multiple dynamic obstacles.

Given an overall obstacle avoidance risk level ϵ with the obstacle avoidance on average in finite time objective, (7.9) with a finite T where the conformal unsafe set becomes $O_t = \bigcup_{j=1}^{n_{\text{obs}}} O_t^j$ where O_t^j are the almost surely unsafe set occupied by obstacle j at time t. In particular, the desire probabilistic obstacle avoidance on average $\frac{1}{T}\sum_{i=t-T+1}^{t} \mathbb{P}[\hat{\boldsymbol{o}}_{i+1} \in O_{i+1}] \leq \epsilon$ for finite T. Using Boole's inequality, the following inequality holds for each i,

$$\mathbb{P}[\hat{\boldsymbol{o}}_{i+1} \in \bigcup_{j=1}^{n_{\text{obs}}} O_{i+1}^j] \leq \sum_{j=1}^{n_{\text{obs}}} \mathbb{P}[\hat{\boldsymbol{o}}_{i+1} \in O_{t+1}^j].$$

Therefore, enforcing the probabilistic state constraint $\mathbb{P}[\boldsymbol{x} \in \mathbb{R}^{n_x} \,|\, \boldsymbol{x}_{t+1} \in O_{t+1|t}^{j,\epsilon,c}] \leq \frac{\epsilon}{n_{obs}} \triangleq \epsilon_n$ for each obstacle at each time instance t, the desired safety requirement is satisfied. Note, by construction of the conformal prediction unsafe set with risk-level ϵ_n for obstacle j as $O_{t+1|t}^{\epsilon_n,c}$, the affine state constraint (7.52) is sufficient to provide safety for each obstacles with a tightened risk-level.

However, consolidating the risk levels does not solve the proposed MPC method (9) having a polynomial growing number of optimization variables and constraints with polynomial growing obstacles. If sufficient training data can be acquired to obtain an obstacle dynamics model, a higher-level trajectory planner like [202] would be more computationally efficient to implement. However, it is important to note that algorithms proposed in [202] will not consider individual feasible sets at each future time instance but the union of all unsafe sets along the prediction horizon. This simplification can lead to infeasibilities in the optimal path planning problem.

Choosing Parameters

In this section, the choice of tuning parameters in the horizon predictor, the construction of the conformal prediction set, and parameters in the PA-MPC problem will be discussed. Qualitative behaviors of the framework will be discussed in this section to provide intuitions. In the later section 7.6, numerical simulations are conducted in the following section to provide quantitative empirical performance to support these discussions.

The accuracy of the horizon predictor heavily depends on the time-embedding length L or equivalently the embedded time window $t_L = L(1/f_{obs})$ sec where f_{obs} is the obstacle position sampling frequency. The prediction formulation uses the time delay as the lifted states to extract a local linear model, a recurrence linear relationship where every next-step prediction depends on the most recent L states. If t_L is too large, the local linearization from pseudo-inverse (7.22) "flatten" nonlinearities within the time window. On the other hand, if t_L is too small which limits the rank of A_t^j in (7.22), the learned update model can be inaccurate ignoring higher-order behaviors.

The planning horizon t_h of the MPC is another important time window parameter where $t_h = n_h/f_c$ where f_c is the controller update frequency. Typically, longer planning horizons t_h are recommended to approximate the optimality and guarantee that infinite horizon planner offers. The upper bound of t_h is typically dictated by the QP solver's ability to compute a solution given planning interval Δt. However, to construct an obstacle avoidance safety constraints (7.52) requires obstacle set predictions window $t_p = n_p/f_{obs} \geq t_h$ along the horizon. The prediction window is the number of recursion (7.27) – (7.28) are computed to make future obstacle forecasts. However, the prediction horizon has the reverse in performance since the recursive forecast in (7.28) will make the heuristic unsafe set grow exponentially in prediction time. Therefore, the least prediction window $t_p = t_h$ is chosen to reduce making overly conservative horizon forecasts which results in overly conservative trajectory planning.

Combining the above observations, the embedded time window t_L should match with the prediction horizon window t_p and the MPC planning horizon t_h, $t_L = t_p = t_h$, which is employed in similar applications ([203]). For all the numerical simulations and hardware validations conducted in this chapter, only the case where $t_p = t_h = t_L$ is studied.

For the adaptive conform prediction algorithm, the learning rate γ heavily dictates the algorithm's ability to adapt to consistent prediction inaccuracy. As observed in [177], the MPC optimization can suffer from infeasibility due to sudden changes in the size of the uncertainty sets when the learning rate γ is too big. But the theoretical coverage under distribution shift in [109] requires $\gamma > 1$ resulting in rapid expansion of the prediction set. For practical implementation, the authors select learning rate $\gamma < 1$ despite slower adaptation to avoid numerical infeasibilities. Further, the heuristic horizon unsafe sets for $k = 2$ to $k = n_h$ are expected to allow the vehicle to make trajectory corrections in advance to reduce the need for an overly conservative conformal unsafe set for the immediate step $k = 1$.

Lastly, a maximum conformal unsafe set radius ρ^{ϵ}_{t+1} in (7.44) on top of the choice of $\gamma < 1$ is strongly recommended. In particular, the conformal radius is upper bounded as $\rho^{\epsilon}_{t+1} = \min\{\rho^{\epsilon}_{t+1}, r^{\Delta t}_{max}\}$, where $r^{\Delta t}_{max}$ is the maximum distance that the ego vehicle can travel for in time t_h given any feasible input and any stationary initial condition. If the obstacle can step-wise traverse exceeding this limit, such obstacle will be inevitable because of torque limits.

7.6 Simulation Results

To assess the empirical performance of the proposed control framework, a series of numerical simulations are conducted with varying parameters and obstacle behaviors. The simulations model the ego vehicle as a point mass system, a simplification chosen for numerical efficiency in representing vehicle dynamics. The ego vehicle is tasked to follow a reference trajectory, denoted as $\boldsymbol{p}^{\text{ref}}$ while avoiding unknown moving obstacles upholding a given risk-tolerance $\epsilon \in (0, 1)$. The ego vehicle's system state is $\boldsymbol{x} = [x, y, z, v^x, v^y, v^z]^T$, consists of its position and linear velocities with respect to a fixed frame. The ego vehicle is controlled by linear acceleration inputs, $\boldsymbol{u} = [a^x, a^y, a^z]^T$, also with respect to the same fixed frame. The discrete-time ego vehicle system dynamics is

$$\boldsymbol{x}_{t+1} = \left(\begin{bmatrix} I_{3\times3} & \Delta t \cdot I_{3\times3} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix}\right) \boldsymbol{x}_t + \begin{bmatrix} 0_{3\times3} \\ \Delta t \cdot I_{3\times3} \end{bmatrix} \boldsymbol{u}_t. \tag{7.58}$$

Suppose the ego vehicle is controlled by the proposed prediction-based MPC (7.55) with a planning rate of 20 Hz ($\Delta t = 0.05\ s$), and the obstacle measurements update at the same frequency. Input limits are set to $\boldsymbol{u}_{max} = [2, 2, 16]^T\ m/s^2$ and $\boldsymbol{u}_{min} = [-2, -2, -6]^T\ m/s^2$. These limits reflect the reduced-order modeling of the quadcopter system in acceleration control mode.

In the nominal scenarios with risk-tolerance $\epsilon = 0.01$ (illustrated in Fig. 7.6), the MPC parameters are as follows: stage costs $Q_k = \text{diag}([50, 50, 50])(k + 1)$ for $i \in \mathbb{Z}^{n_h-1}_{k=1}$, input cost $R = \text{diag}([0.1, 0.1, 0.01])$, and terminal stage cost $Q_f = \text{diag}([50, 50, 50])n_h$. The MPC horizon is set at 0.5 s with a time step of 0.05 s. To mitigate numerical infeasibilities and overly conservative predictions, obstacle avoidance constraints for future time horizons from $k = 2$ to $k = n_h$ are relaxed, as per formulation in (7.55). Slack variable costs are set as $\lambda_{0,k} = 50(2^{n_h-k})$ and $\lambda_{1,k} = 20$, prioritizing trajectory planning that avoids the heuristic unsafe set in the near term while relaxing constraints for future time instances. For the sequential QP formulation, a finite termination criterion of $\omega_{max} = 5$ iterations is employed, with trust region parameters selected as $m_x = 100$, $m_u = 50$, $\tau_x = 0.5$, and $\tau_u = 0.5$.

An obstacle, following the dynamics of a Frisbee as detailed in [167], is introduced with randomly selected initial conditions. To evaluate the framework's efficacy, all obstacle trajectories are synthesized and adjusted with respect to the ego vehicle's reference trajectory to induce guaranteed collision scenarios if the vehicle follows the desired reference closely. Collision timings are randomly set between 12 and 25 seconds, challenging the framework's capability to varying observation windows. For real-time obstacle avoidance, only the current time instance's obstacle positions in x, y, z are available to the horizon predictor, with an added Gaussian noise component sampled from $\mathcal{N}(0, 0.05)$ for each x, y, and z measurements.

The parameters selected for the horizon predictors include an embedding length of $L = 10$ and a sample size of $N = 100$ for constructing the Hankel and Page matrix. The extended Kalman filter, as delineated from (7.27) to (7.35), is utilized for constructing the heuristic unsafe set occupied by the obstacle. This filter is initialized with a heuristic process noise $\mathcal{W}_0 = 0.1 I_{L\times L}$ and a measurement noise covariance $\mathcal{V}_0 = 0.05$ for each x, y, z observables which will be refined based on the noise estimation (7.20) after reaching data window threshold N. To obtain the conformal unsafe set, a range of learning rates

$$\gamma = \{0.0008, 0.0015, 0.003, 0.009, 0.17, 0.03, 0.05, 0.08, 0.13\}$$

are employed, selecting the optimal rate at each time t following the fully adaptive conformal algorithm introduced in Chapters 2 and 6. The finite-time averaging window size T_{safe} is set at 20.

The nominal case results, depicted in (7.6), reveal that despite the varying conservatism given different risk levels given identical obstacle scenarios and all other parameters. Despite the mean prediction, denoted as the red solid lines, are not aligned with the actual obstacle's true path in the future (orange color dotted line), the unsafe "cone" formed by the prediction unsafe set at future time instances sufficiently captures the obstacle's future path. The ego vehicle, by enforcing obstacle avoidance constraints along the MPC horizon, successfully achieves the desired probabilistic obstacle avoidance (7.8). This is noteworthy, considering that the MPC program (7.55) theoretically guarantees only average probabilistic finite-time obstacle avoidance. This safety enhancement is a result of the heuristic unsafe set

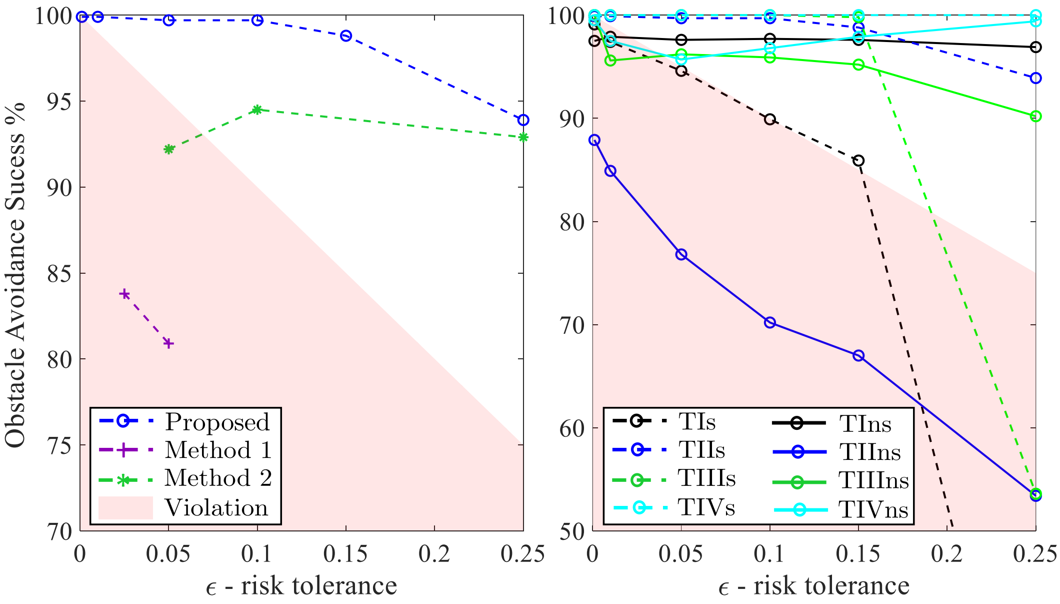

Figure 7.5: Simulation Result and Comparison. [Left] Simulation with agent dynamics (7.58) and a Frisbee obstacle with dynamics adopted from [167]. Result comparison against prior works for performance comparisons over different risk levels. Method 1 is the Uncertainty-Informed Model Predictive Control presented in Chapter 6. Method 2 is the Singular Spectrum Analysis Model Predictive Controller presented in Chapter 5. [Right] Simulation performance of the proposed framework against obstacles described in table 7.1 where the legends "TIs" stands for type I obstacle with MPC slacking for the heuristic unsafe sets and "TIVns" stands for type IV obstacle without slacking in the MPC formulation.

integrated along the horizon, which enables the ego vehicle to deviate from the reference trajectory in the presence of potential future collisions. However, the heuristic unsafe set, despite its practical effectiveness, does require extensive and possible invalid assumptions to provide equivalent probabilistic obstacle avoidance guarantees.

Additionally, Monte-Carlo (MC) simulations were conducted to statistically evaluate the proposed framework's performance relative to various scenarios: (1) obstacle behaviors, (2) obstacle position measurement noise characteristics, (3) planning horizon window, (4) the impact of slack variables, and (5) the choice of risk-level and overall framework efficacy. The prediction-aware MPC (7.55) was solved using Gurobi [173] on an Intel i7-9700K CPU @3.6 GHz. Both the dynamic simulator and the proposed framework were implemented in MATLAB.

The performance of the proposed prediction-based MPC is compared with prior works, presented in Chapters 5 and 6. Recall Chapter 5 uses a singular spectrum analysis-based linear prediction paired with a distributionally robust approach to

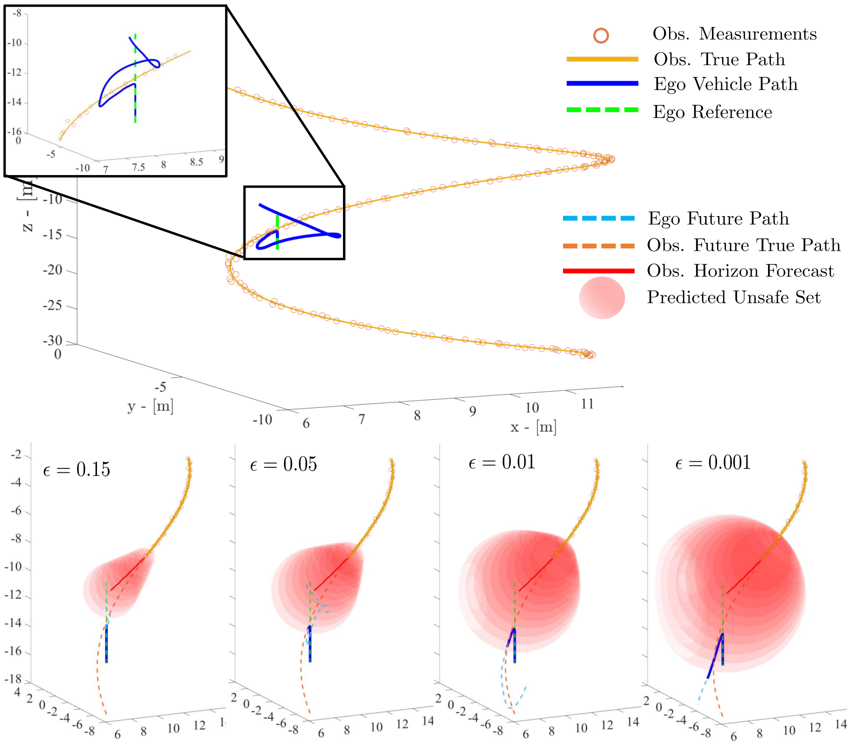

Figure 7.6: Simulation with agent dynamics (7.58) and a Frisbee obstacle with dynamics adopted from [167]. The same obstacle behaviors are simulated while the agent tracks the same "Figure 8" reference trajectory with four risk levels $\epsilon = \{0.15, 0.05, 0.01, 0.001\}$. The simulation is designed to be difficult: the vehicle must deviate from its reference trajectory as the obstacle trajectory is designed to intersect the agent's reference trajectory with noisy obstacle position measurements.

make prediction uncertainty quantification. The method represented in Chapter 6 uses the same linear prediction introduced in Chapter 5 and designs an adaptive conformal prediction approach to quantify prediction uncertainty. The simulation details can be found in Chapter 5 where the same obstacle (Frisbee) behaviors are simulated with an ego vehicle with matching dynamics and input limits. The empirical obstacle avoidance statistics are plotted in Fig. 7.5 [Left]. The results show the proposed method outperforms the past works across a wide range of risk-tolerances and can truly uphold the ϵ risk-tolerance as the rate of failure to avoid obstacles.

The right subplot of Fig. 7.5 summarizes the obstacle avoidance performance of the proposed method using model (7.58) against the four differently behaving obstacles described in Table 7.1. Based on the tabulated result, the proposed method when slacked variables are incorporated (all dotted results) can uphold the desired probabilistic obstacle avoidance specification at low-risk levels ($\epsilon < 0.15$).

I	Indifferent Const. Spd.	$\boldsymbol{o}_{t+1} = \boldsymbol{o}_t + \Delta t \boldsymbol{v}^o$	$\boldsymbol{o}_0$ uniformly sampled, $\boldsymbol{v}^0$ uniformly sampled
II	Indifferent Frisbee	$\boldsymbol{o}_{t+1} = \boldsymbol{o}_t + \Delta t f^d(\boldsymbol{o}_t + \zeta_t^o)$ ([167])	Uniform sample of initial pitch angle, launching speed, flight path angle, and initial spin rate.
III	Indifferent Figure 8	$\boldsymbol{p}_t^0 = \begin{bmatrix} d_x^o cos(t) & d_y^o sin(2t) & d_z \end{bmatrix}^T$	Parameters d_x^o and d_y^o dictates the geometry of the figure "8" trajectory which is uniformly sampled between $[0.5, 1.5]$. The constant parameter d_z determines the height of the obstacle trajectory.
IV	Adversarial chaser	$\boldsymbol{o}_{t+1} = \boldsymbol{o}_t - \mathbf{1}_{k \leq 0.3} \Delta t P_{gain}(\boldsymbol{o}_t - \boldsymbol{p}_t)$	The chaser is guided by a proportional controller where its P_{gain} is uniformly sampled from $[0.1, 1.0]$. Parameter k is randomly sampled from $[0, 1]$ that determines whether the chaser chases or stops.

Table 7.1: Summary of simulated obstacle and their behaviors.

Obstacle Behaviors

To evaluate the framework's capability to navigate around obstacles with unknown intentions and differential behaviors, Monte Carlo (MC) simulations are conducted where the planner interacts with four distinct types of obstacles. Each obstacle type, characterized by unique dynamics and behavior, was introduced once per simulation run. The planner's success rate in avoiding these adversarial obstacles is presented in Figure 7.5. These obstacle dynamics, along with their randomized parameters and initial conditions, are detailed in Table 7.1 with additional performance metrics.

For a comprehensive analysis, there are 1000 MC simulations conducted for each parameter combination. The primary metrics for assessment were the success rate in obstacle avoidance and the conservativeness of the planner. The latter was quantified by measuring the minimum distance maintained between the center of the obstacle and the agent. It's important to note that all parameters remained the same across different simulations, except the obstacle dynamics.

The results of these simulations are collated in Table 7.5. These results depict how the ego vehicle successfully navigated around each of the four differently behaving obstacles. Additionally, Fig. 7.6 presents an illustrative example of a simulated trajectory. This figure highlights the predicted unsafe sets for the ego vehicle under varying levels of risk tolerance, demonstrating the robustness of the planning framework given intention, behavior unknown dynamic obstacles.

Noise type and Intensity

Central to the efficacy of the denoising steps, described in Section 7.2 is the assumption that the additive noise adheres to a Gaussian distribution with a zero mean. To test this assumption, additional MC simulations are conducted where the planner navigates around a Frisbee obstacle, adhering to all nominal case parameters except for variations in noise level (standard deviation) and noise type. This investigation initially focused on assessing the horizon predictor's resilience against varying intensities of measurement noise. These noises were simulated as Gaussian distributions with standard deviations $0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.4\, m$. The framework maintains the designated risk level ($\epsilon = 0.05$) effectively for noise standard deviations up to $0.15\ m$ (Tabulated results can be found in Table 7.2). However, when noise standard deviation surpasses $0.2\, m$, there is a marked decline in planner performance. This deterioration is attributed to inadequate separation between signal and noise, leading to compromised predictor accuracy. Conversely, at lower noise levels, more conservative obstacle-avoiding behaviors are observed. This is quantified by a greater minimum distance (d_{min}) both planned and actualized between the obstacle and the agent. This counterintuitive phenomenon stems from poor conditioning in the pseudoinverse (see Equation (7.22)), which catalyzes over-fitting and instability in the reduction of the data-driven transition matrix. [6]

Horizon Window

The planning horizon window, denoted as $t_h = t_p = t_L$, is pivotal in determining the efficacy of the proposed framework. Based on the Monte-Carlo simulation results in Table 7.4, longer horizon windows are observed to enhance performance, particularly when slack variables are introduced. While no-slack scenarios exhibit high success rates and small minimum distances (d_{min}), they also present a significant risk of infeasibility. Such infeasibilities are critical as they render the probabilistic guarantees of Equation (7.55) void when a previous iteration roll-out is issued, potentially leading to unsafe trajectories. In contrast, when slacking is incorporated,

[6] To resolve this issue, controlled additive noise is recommended, particularly in scenarios where the noise level is uncertain. This approach should improve the condition number of the pseudoinverse. Alternatively, reducing slack variable costs or incrementally increasing stage and terminal costs as i progresses within the range $\mathbb{Z}_1^{n_h-1}$ could effectively balance reference tracking with future collision risks. Additionally, while reducing the embedding length L might mitigate over-fitting, care must be taken to avoid inadequate lifting and consequently a poor obstacle.

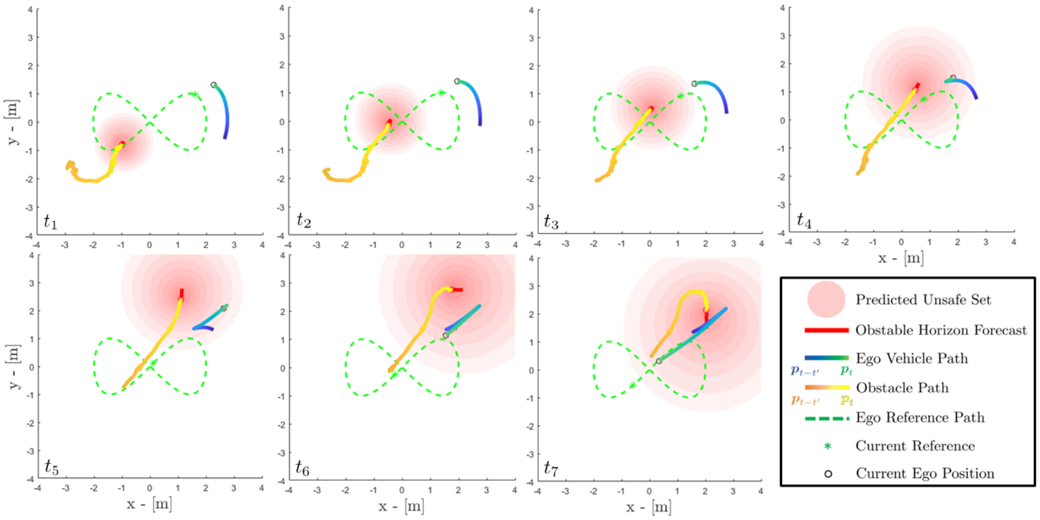

Figure 7.7: Hardware result of UGV following a desired “Figure 8" reference trajectory while avoiding a human-driven adversarial obstacle. The obstacle objective is to intersect the vehicle.

the performance shows a decrement, with d_{min} increasing exponentially with the horizon length. Additionally, the computational time, encompassing both the MPC planner and the horizon predictor, scales in polynomial time with the horizon length, which must be taken into consideration for real-time applications.

Slack variable

The studies on the horizon window indicate that slacking uniformly mitigates the risk of infeasibilities (see Table 7.4), resolving conflicts among obstacle avoidance constraints, torque bounds, and trust region constraints. Through empirical analysis, the choice slack variable costs $\lambda_{0,i}$ and $\lambda_{1,i}$ are sampled, for $k \in \mathbb{Z}_1^{n_h-1}$. The simulation results, tabulated in Table 7.3, reveal a trade-off between the obstacle avoidance success rate and the collision-avoiding distance where increasing slack cost asymptotically approaches the unslacked behaviors. The choice of slack variables contributes most in balancing between maintaining safe distances and adhering to desired trajectories.

Risk-level versus Tracking Performance

To elucidate the relationship between the planner’s performance with the chosen risk level, extensive empirical data is compiled in Table 7.5, outlining the performance across four different obstacles at various risk levels. The empirical results show that applying slacking at a low-risk level produces the highest obstacle avoidance success rates. However, this risk aversion comes with its costs, notably an increase in numerical infeasibility rates and larger d_{min} values, which implies a reduction in reference tracking accuracy.

Despite the risk level being directly factored into the construction of probabilistic unsafe sets occupied by obstacles, the framework's performance at small risk levels ($\epsilon < 0.1$), coupled with slacking, surpasses the theoretical guarantees of average probabilistic obstacle avoidance in finite-time. This enhancement in performance is attributable to the incorporation of heuristic unsafe sets throughout the horizon, as demonstrated in Table 7.4, and the systematic use of sequential conservative convexifications as formulated in Eqn. (7.52).

7.7 Hardware Validation: UGV

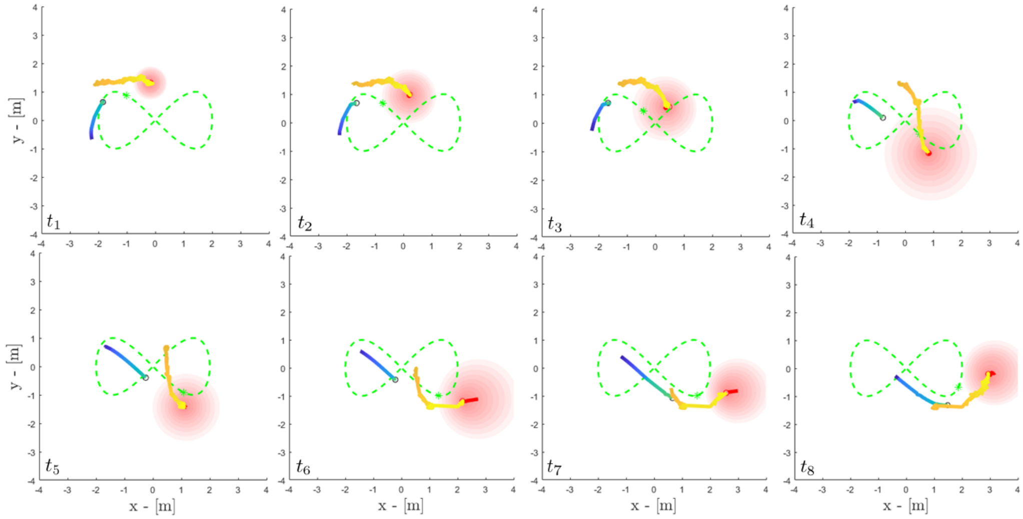

Figure 7.8: Hardware Validation Result of UGV following a desired "Figure 8" reference trajectory while avoiding a human-driven adversarial obstacle. The obstacle objective is to block the vehicle from its target path without directly intersecting the obstacle.

The proposed framework is tested on hardware, summarized in Fig.7.1, on two platforms: (1) a ground vehicle (UGV), and (2) a hexacopter drone (UAV) as the ego vehicle. Both vehicles are tasked to track a reference "Figure 8" trajectory while avoiding obstacles. There are differently behaving obstacles, guided by humans, are introduced individually and in combinations. See supplementary video.

Indoor experiments are conducted for both platforms under an Optitrack Motion Camera to provide position ground true to the ego vehicle as well as the obstacles. To verify the proposed framework's robustness to measurement error, noises are manually added into the obstacle position during all experiments. For UGV tests, 2D rigid body obstacles are considered where 2D obstacle position with respect to an inertial frame is streamed to the ego vehicle at 20 Hz frequency. The injected noise has zero mean and standard deviation of 0.05 m, and all obstacles have a radius of less than 0.2 m. For UAV tests, 3D rigid body obstacles are considered, similarly with positions streamed to the ego vehicle at 20 Hz. There are white noises with a standard deviation of 0.05 m for each coordinate position added on top of the motion-captured obstacle positions, mimicking obstacle detecting and tracking errors using real sensors.

The quadratic program (7.55) is run onboard for both platforms using the open source solver OSQP [204] with OSQP Eigen C++ interface.

UGV Details

The Army GVR-BOT, differential-drive, is used as the unmanned ground vehicle platform. For onboard computing, the vehicle is powered by a Jetson AGX Orin computer running the proposed framework in a ROS environment. The UGV accepts linear velocity commands with respect to the body x axis, denoted as v, and angular velocity commands with respect to the body z axis, denoted as ω. The following discrete-time kinematic model is used to describe the vehicle dynamics (the Euler discretization of the UGV model used for differential-driven vehicles in Chapter 4):

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} \cos(\theta_t) & 0 \\ \sin(\theta_t) & 0 \\ 0 & 1 \end{bmatrix} \Delta T \underbrace{\begin{bmatrix} v_t \\ \omega_t \end{bmatrix}}_{\boldsymbol{u}_t}. \tag{7.59}$$

With system states $\boldsymbol{x} = [x, y, \theta]^T$ being x, y position, and heading angle. Since the proposed Prediction-Based MPC (7.55) needs to be solved sequentially to reach the tightest obstacle convexification, sequential linearization of (7.59) around its previous iteration optimal solution is performed, following equations (7.3)-(7.4). More information on sequential convex programming can be found in 2. With a controlling frequency of 20 Hz and the input range $v \in [-1.0, 1.0]$ m/s and $\omega \in [-1.4, 1.4]$ rad/s, a conservative single-step tracking error of 0.1 m is employed, a heuristic safety buffer, due to linearization and unmodeled model residuals. Such buffer will be incorporated on top of the radius of the predicted unsafe set of each obstacle. The vehicle is tasked to track a periodic “Figure 8” trajectory, designed without considering obstacles, which is kinodynamically feasible given the input limits. The dynamic obstacles, driven by human researchers to act as the intention unknown obstacles, are introduced while the ego vehicle is tracking the given reference. Only a subset of the experimental results is provided in Fig. 7.7-7.9 to showcase the proposed planner's performance on hardware.

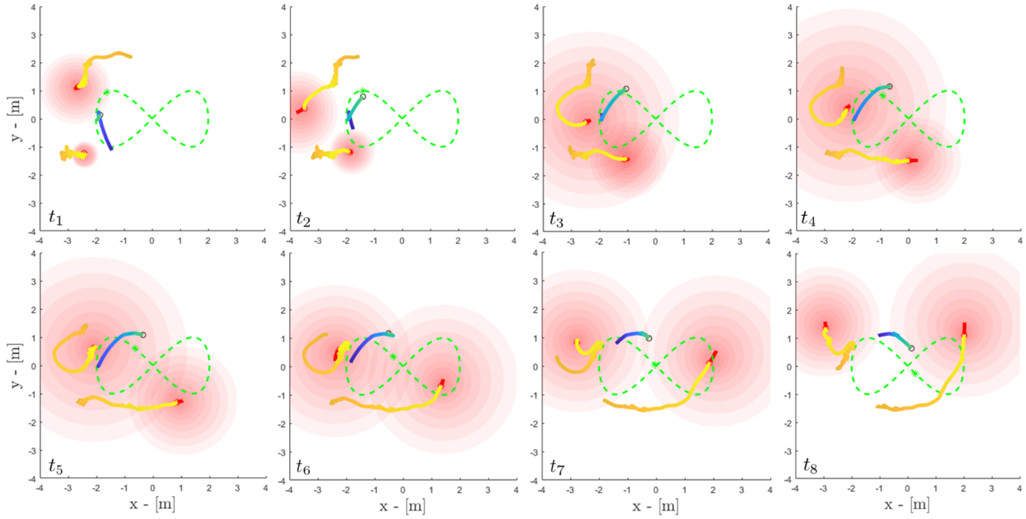

Figure 7.9: Hardware result of UGV following a desired "Figure 8" reference trajectory while avoiding two human-driven adversarial obstacles. The obstacle's objectives are to intersect the vehicle or block its current path.

Figure 7.7 consists of time snapshots of the trajectories of both the ego vehicle and an adversarial obstacle aiming for a head-on collision. Using a predictor horizon of 0.5 s and a risk tolerance of 0.01, the ego vehicle successfully dodges the obstacle and returns to reference tracking. Using the predicted unsafe set, the ego vehicle can make trajectory adjustments before the collision is immediate and optimizes reference tracking once the predicted unsafe set no longer poses collision threats.

Figure 7.8 depicts the proposed planner's result given a potentially friendly obstacle that hovers around its goal points. Given a completely different obstacle behavior and intent, the proposed planner can successfully mitigate the obstacle and follow the reference target tightly. With a risk tolerance of 0.1, the predicted unsafe set is noticeably less conservative. Nevertheless, the ego vehicle guided by the proposed planner keeps a safe distance from the obstacle without any heuristic tuning.

To showcase the planner's performance given multiple obstacles, two dynamic obstacles are introduced simultaneously with unspecified behavior in Fig. 7.9. With a risk-averse $\epsilon = 0.01$ preference, the ego vehicle is still able to dodge aggressive obstacles without collisions. As provided in the trajectory snapshots, the obstacles are moving significantly faster with a top speed of 2 m/s exceeding the ego vehicle's maximum velocity capability. Benefiting from the rather conservative predicted unsafe set, the vehicle can plan trajectories to escape without immediate collision.

Nevertheless, if the obstacle kept chasing the ego vehicle maintaining that top speed would still lead to an inevitable collision.

7.8 Hardware Validation: UAV

A custom hexacopter is used to validate the obstacle avoidance planner in 3D settings. This hexacopter is equipped with a single-board companion computer powered by an AMD Ryzen 5 3550H processor. A Hex Cube Orange flight controller runs PX4 autopilot [148] to provide a low-level feedback tracking controller for the hexacopter. A safeguard tether attaches to the multirotor base and a ~100 g passive weight eliminates slack which has minimal interference to flight behavior. All flight tests are conducted in an indoor arena equipped with an Optitrack motion capture system that samples rotor-craft pose at 50 Hz and obstacle positions at 20 Hz.

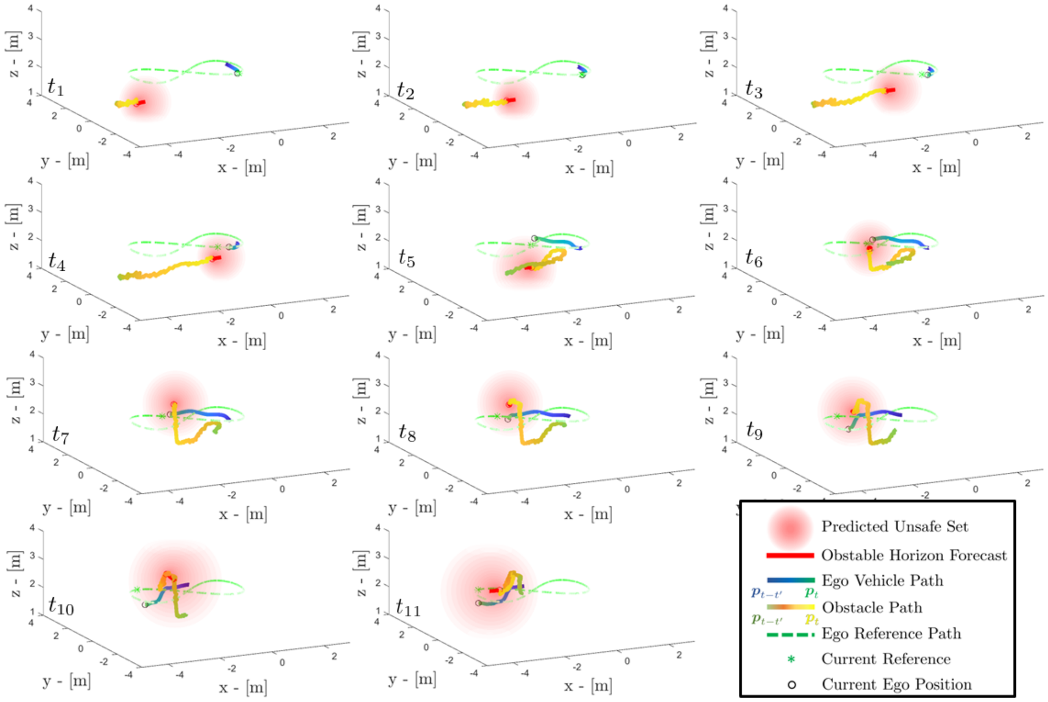

Figure 7.10: Hardware result of UAV following a desired "Figure 8" reference trajectory while avoiding human-driven adversarial obstacles. The obstacle objective is to intersect with the vehicle multiple times at random instances.

A single integrator model is used as the UAV's reduced order model to plan obstacle-free trajectories given 50 Hz measurement feedback. With a constant desired yaw angle, the control inputs to the systems are inertial frame linear velocities in x, y, and z directions which are then tracked by the onboard autopilot at Cascaded rates. As implementation details, the time window parameters are selected as $t_h = t_p = 0.6\ s$, where the MPC horizon $n_h = 30$ and the predictor horizon $n_p = 12$. The heuristic obstacle avoidance unsafe sets are obtained at the MPC planning rates by linearly interpolating between the Gaussian distributions (7.36) with matching timings as the MPC planner. The heuristic unsafe sets are then constructed following (7.53).

With appropriate tuning and an input range of $[-1, -1, -0.5]^T$ to $[1, 1, 1]^T$ m/s, a maximum step-wise position tracking error at 50 Hz is 0.10 m with a 99.5% confidence. To account for the tracking uncertainty as well as the vehicle geometry, the safety radii $\rho^{\bar{\epsilon}}_{t+1}$ in Eqn. (7.56) and r^{ϵ}_{t+k} in (7.57) for $k = \{2, \cdots, n_h\}$ are increased by a maximum step-wise tracking error (for more details, see Chapter 4) and the vehicle radius, which are summed to be 0.2 m. The ego vehicle is tasked to track a “Figure-8" reference trajectory in the $x - y$ plane at a constant height of 2 m.

The proposed framework showed promising results of a custom UAV avoiding human-guided obstacles that approach the ego vehicle at random times while performing reference tracking. As shown in figure 7.10, the experiment results in different time snapshots showcase the framework’s ability to avoid moving obstacles with unknown intentions. With a risk level of 0.05, the ego vehicle guided by the proposed planning can also successfully avoid a 3D obstacle approaching the ego vehicle in various directions. Despite the noisy obstacle trajectory and rather unpredictable behavior, the proposed framework is still able to make relatively accurate horizon forecasts. Nevertheless, the prediction uncertainty robustness provided by the predicted unsafe set is sufficient to inform the planner to make path corrections accordingly.

As the main takeaway from these experimental results, the proposed planner is dynamics-, dimension-, and behavior-agnostic while offering risk-tolerance intuitive tunability. For low-speed obstacles, the proposed planner behaves similarly to the standard potential field method with a rigorously sized repulsive field. For high-speed obstacles, the proposed planner provides unsafe set previews that are crucial for finding collision-avoiding maneuvers.

7.9 Results and Discussion

In this chapter, a novel obstacle avoidance framework was formulated using a data-driven predicted unsafe set for obstacles with unknown dynamics and intentions. The proposed framework is heuristic-free with intuitive risk-based tunability tailored for hardware-orientated, real-time feasible applications. Constrained by the limited, real-time obtained obstacle position measurement, an online obstacle model identification technique is synthesized using ideas from Koopman and dynamic mode decomposition. Instead of relying on the identified model to make perfect behavior forecasts, the prediction uncertainty is quantified using filtering and statistical inference techniques. Lastly, an efficient sequential QP-based MPC with a deterministic obstacle avoidance constraint is constructed based on the obstacle forecasts and user-chosen risk tolerance.

The proposed framework showed empirical successes in avoiding high-speed, adversarial obstacles with randomized scenarios. The proposed framework has also been validated regarding its sensitivity to measurement noise, showing its robustness against low-intensity noise. The proposed framework is also obstacle behavior and intention agnostic as validated in simulation and hardware. Validation experiments are conducted on hardware for both ground and aerial platforms to demonstrate framework versatility. For both scenarios, the ego vehicle successfully dodges intention-unknown human-guided obstacles while optimizing for its tracking tasks.

Limitations and Future Work

On the note of stochastic uncertainty, the current MPC formulation does not consider any state and process noise nor the use of a feedback policy. The maximum tracking error must be quantified from data, as presented in Chapter 4, to provide probabilistic safe guarantees in the presence of model uncertainties. On the same note, the method presented in Chapter 4 investigated finding probabilistic maximum tracking discrepancy upper bound from data as well as a pairing feedback policy with stability and tracking guarantees. Such tracking discrepancies can be used to construct safe sets that account for both model discrepancies and dynamic obstacles which is a natural extension of this work.

This framework can also be extended to use sample-based planners like model predictive path integral (MPPI), see Chapters 2 and 4, to relax the conservative convexification in reformulating the obstacle avoidance constraint and sequential linearization of the system dynamics. Extending the hardware results in a non-laboratory environment is the immediate future work. In particular, using onboard sensors to localize the ego vehicle, incorporating measurement feedback, and most importantly identifying and localizing dynamic rigid body obstacles are important features for achieving real-world moving obstacle avoidance.

Type	Parameters	0.001	0.01	0.05	0.10	0.2	0.25	0.4
Uniform Noise	%Succ.	99.9	99.9	99.9	99.8	99.7	99.3	82
	d_{min} actual	4.76 ± 4.14	4.29 ± 0.51	2.27 ± 2.61	1.01 ± 0.22	0.96 ± 3.70	0.44 ± 0.16	0.48 ± 0.18
	d_{min} planned	4.67 ± 3.24	4.29 ± 0.51	2.27 ± 0.61	1.02 ± 0.21	0.77 ± 0.69	0.45 ± 0.16	0.51 ± 0.24
	Infeas. %	0.3	0.0	1.33	2.68	29.33	2.4	11.43
Gaussian Noise	%Succ.	99.9	99.9	99.7	99.5	92.9	72.9	2.8
	d_{min}	4.22 ± 2.50	4.09 ± 0.70	1.16 ± 0.28	0.71 ± 0.12	0.56 ± 0.15	0.37 ± 0.16	0.39 ± 0.25
	d_{min} planned	4.14 ± 1.34	4.09 ± 0.69	1.19 ± 0.24	0.75 ± 0.17	0.57 ± 0.16	0.69 ± 0.44	0.39 ± 0.25
	Infeas. %	0	1.67	18.96	20.81	7.43	63.92	0

Table 7.2: Monte Carlo Simulation result of ego vehicle (7.58) avoiding a single indifferent Frisbee Obstacle with position measurement corrupted by unbiased uniform noise and unbiased Gaussian noise.

Slack Cost	(% nominal)	0.05	0.1	0.5	1.0	2.0	5.0	∞
Type II	%Succ.	39.2	59.0	99.5	99.7	99.7	99.9	76.8
	d_{min} actual	0.39 ± 4.30	0.33± 3.31	0.75± 0.21	1.16 ± 0.28	2.74 ± 1.49	2.18 ± 0.68	0.25 ± 0.06
	d_{min} planned	0.66 ± 2.49	0.58 ± 1.10	0.85 ± 0.20	1.19 ± 0.24	2.58 ± 0.17	2.18 ± 0.68	1.10 ± 0.64
	Infeas. %	92.1	85.3	60.4	18.96	16.7	0.1	100

Table 7.3: Monte Carlo Simulation result of ego vehicle (7.58) avoiding a single indifferent Frisbee Obstacle with position measurement with different levels slack costs.

	Horizon (sec)	0.1	0.25	0.5	0.7	1.0	1.5
Yes slack	%Succ.	0.1	42.4	99.7	99.9	99.5	99.6
	d_{min} actual	0.19 ± 0.03	0.19 ± 0.04	1.16 ± 0.28	3.34 ± 3.21	6.01 ± 5.69	9.90 ± 2.45
	d_{min} planned	1.38 ± 1.65	0.50 ± 0.25	1.19 ± 0.24	3.24 ± 0.62	5.79 ± 2.49	9.87 ± 2.50
	Infeas. %	60	75	18.96	0.3	0	0
No Slack	%Succ.	0.4	1.5	76.8	98.1	99.5	99.6
	d_{min} actual	0.49 ± 0.48	0.17 ± 0.06	0.25 ± 0.06	0.53 ± 0.16	1.78 ± 0.64	0.88 ± 0.21
	d_{min} planned	0.93 ± 0.77	0.62 ± 0.40	1.10 ± 0.64	1.06 ± 0.64	1.87 ± 0.75	0.9 ± 0.19
	Infeas. %	75	93.33	100	95.31	17.69	28.94

Table 7.4: Monte Carlo Simulation result of ego vehicle (7.58) avoiding a single indifferent Frisbee Obstacle with position measurement with different matching prediction and planning horizon.

Cases	Methods	$C^*_{\epsilon=0.001}$	$C^*_{\epsilon=0.01}$	$C^*_{\epsilon=0.05}$	$C^*_{\epsilon=0.1}$	$C^*_{\epsilon=0.15}$	$C^*_{\epsilon=0.25}$
Type	%Succ.	99.1	97.4	94.6	89.9	85.9	19
I	d^a_{min}	1.50 ± 1.93	0.96 ± 1.98	0.73 ± 1.98	0.72 ± 2.20	0.75± 2.71	1.12 ± 2.73
Yes	d^p_{min}	1.44 ± 1.26	0.93 ± 1.27	0.76 ± 1.25	0.76 ± 1.47	0.75 ± 1.44	0.99 ± 1.86
slack	Infeas. %	33.8	56.16	85.52	91.77	95.34	63.68
Type	%Succ.	97.5	97.9	97.6	97.7	97.6	96.9
I	d^a_{min}	0.78 ±0.57	0.70 ± 0.43	0.67 ± 0.29	0.64 ± 0.22	0.60 ± 0.17	0.57 ± 0.14
No	d^p_{min}	1.76 ± 0.30	1.36 ± 0.29	0.98 ± 0.23	0.81 ± 0.22	0.71 ± 0.17	0.67 ± 0.15
slack	Infeas. %	84.31	82.94	80.53	78.4	77.56	90.51
Type	%Succ.	99.9	99.9	99.7	99.7	98.8	93.9
II	d_{min} actual	5.25 ± 2.79	2.75 ± 2.53	1.16 ± 0.28	0.71 ± 0.17	0.45 ± 0.11	0.374 ± 3.95
Yes	d_{min} planned	5.18 ± 1.22	2.75 ± 2.52	1.19 ± 0.24	0.82 ± 0.18	0.65 ± 0.21	0.57 ± 0.92
slack	Infeas. %	0	1.5	18.96	56.97	73.68	80.72
Type	%Succ.	87.9	84.9	76.8	70.2	67	53.4
II	d^a_{min}	0.34 ± 0.10	0.29 ± 0.08	0.25 ± 0.06	0.24 ± 0.05	0.23 ± 0.05	0.21 ± 0.04
No	d^p_{min}	5.73 ± 1.40	3.39 ± 1.15	1.10 ± 0.64	0.53 ± 0.37	0.46 ± 0.23	0.45 ± 0.21
slack	Infeas. %	100	100	100	93.87	82.7	75.28
Type	%Succ.	100	100	100	100	99.8	53.6
III	d^a_{min}	1.99 ± 0.46	1.47 ± 0.59	0.94 ± 0.65	0.67 ± 0.67	0.49 ± 0.65	0.41 ± 0.95
Yes	d^p_{min}	1.98 ± 0.39	1.45 ± 0.40	0.92 ± 0.45	0.65 ± 0.50	0.48 ± 0.36	0.44 ± 0.68
Slack	Infeas. %	0	0	0	2.1	13.63	78.54
Type	%Succ.	100	95.6	96.2	95.9	95.2	90.2
III	d^a_{min}	1.88 ± 0.28	1.47 ± 0.72	1.09 ± 0.46	0.84 ± 0.30	0.67 ± 0.20	0.41 ± 0.10
No	d^p_{min}	0.66 ± 0.17	0.67 ± 0.84	0.93 ± 0.76	0.82 ± 0.53	0.69 ± 0.37	0.44 ± 0.19
Slack	Infeas. %	11.43	13.49	29.52	30.34	28.26	31.82
Type	%Succ.	100	100	100	100	100	100
IV	d^a_{min}	2.31 ± 1.12	1.80 ± 1.73	1.18 ± 1.57	0.92 ± 1.63	0.80 ± 1.80	0.44 ± 0.54
Yes	d^p_{min}	2.31 ± 1.34	1.80 ± 1.73	1.12 ± 1.57	0.93 ± 1.63	0.82 ± 1.79	0.55 ± 0.36
Slack	Infeas. %	0	0	1.2	5.4	10.2	36.4
Type	%Succ.	99.5	97.9	95.7	96.8	97.9	99.4
IV	d^a_{min}	0.38 ± 0.14	0.34 ± 0.11	0.30 ± 0.08	0.29 ± 0.07	0.30 ± 0.08	0.32 ± 0.07
No	d^p_{min}	4.71 ± 0.29	3.85 ± 0.60	2.61 ± 0.59	1.93 ± 0.47	1.49 ± 0.38	0.84 ± 0.24
Slack	Infeas. %	99.9	99.9	100	100	100	98.29

Table 7.5: Summary of results from MC simulations. Each block is generated with 1000 Monte Carlo simulations of the same agent tracking the same reference trajectory avoiding a single obstacle governed by the same type of dynamics described in Table 7.1. The empirical statistics are generated with increasing obstacle velocities to emphasize the proposed methods' ability to uphold the user-chosen risk factor ϵ.

Chapter 8

SUMMARY

Safety is a critical requirement for the autonomous exploration of complex environments characterized by unknown, unstructured, and dynamic uncertainties. This thesis has presented a novel planning and control framework for robotic autonomy, offering probabilistic safety guarantees in such environments. The core elements of this framework include data-driven model identification, uncertainty quantification of model prediction accuracy, and the transformation of risk-aware chance constraints into deterministic state constraints for optimization.

Ensuring system stability and safety in the presence of these uncertainties necessitates robust vehicle planning and control strategies. Moreover, understanding the impact of uncertainties on closed-loop system safety and overall performance is crucial. Addressing these challenges, my research has established a comprehensive framework for safety, answering these questions posted in Chapter 1:

Q1 How to quantify uncertainty arising from model mismatches?

A1: Data-Driven Uncertainty Quantification from closed-loop tracking performance.

This thesis contained two possible solution methods: an online method through learning the key design parameter Surface-at-Risk using Gaussian Process Regress, as discussed in Chapter 3, and an offline method through identifying matched and unmatched model discrepancy upper bounds given a user-specified confidence using conformal predictions, detailed in Chapter 4.

Q2 What is the impact of unstructured uncertainties on system performance and stability?

A2: Uncertainty Aware Controller Augmentation

Building on the discrepancy upper bounds, it is crucial to robustify the nominal controller, designed based on a reduced-order model, to guarantee stabilizability and safety. This is discussed in depth in Chapter 4. The nominal controller is augmented using an input-to-state stabilizability approach, extending asymptotic stability to the perturbed system. To ensure safety, I analyze and bound the augmented controller's maximum trajectory tracking deviation, creating a "safety tube" within which the system can operate safely, despite model mismatches.

Q3 What methods can be employed to predict dynamic obstacles' future trajectories?

A3: Data-Driven Dynamical Obstacle Identification

The proposed methods focus on learning simple obstacle models in real time which is then analyzed for prediction uncertainty to provide probabilistic guarantees. Two approaches are covered in this thesis for extracting dynamic models of moving obstacles with real-time obtained position-only measurements corrupted by Gaussian white noise.

Chapter 5 employs Singular Spectrum Analysis (SSA), assuming that linear recurrence relationships can describe obstacle trajectories. Chapter 7 proposes a Koopman-inspired approach to learn linear-time varying stochastic models for each unknown dynamic obstacle, without making assumptions about their trajectories, dynamics, intentions, or behavior patterns. Both methods involve time-delay embedding of past position measurements and forward-propagating the identified model for generating obstacle predictions.

Q4 How to reason about prediction accuracy given a probabilistic risk tolerance?

A4: Unsafe Set predictions occupied by dynamic obstacles

I proposed three methods with various fidelity and assumptions for quantifying prediction uncertainty upholding the given risk tolerance. A heuristic method through statistical bootstrapping generates empirical unsafe sets from multiple obstacle models using Singular Spectrum Analysis, as discussed in Chapter 5. I also synthesized a data-driven Kalman Filter for obstacle position estimations using the Koopman-inspired model in Chapter 7. Building on these empirical successes, a rigorous unsafe set quantification technique is proposed using adaptive conformal prediction in Chapters 6 and 7, calibrating heuristic unsafe sets to the given risk-tolerance.

Q5 How to synthesize planners and controllers, considering quantified uncertainties and models, to ensure probabilistic safety and stability for robot autonomy?

A5: Safety-Critical Planning with Probabilistic Guarantees

> To balance safety constraints with the planner's objective function, model predictive planning methods like model predictive control (MPC) and model predictive path integral (MPPI) are selected. Chapters 5, 6, and 7 propose MPC with provable guarantees for probabilistic obstacle avoidance by reformulating the predicted unsafe set as deterministic inequality state constraints. This reformulation allows for the development of a real-time feasible sequential quadratic program while maintaining a user-defined risk level leveraging convex optimization. Leveraging MPPI, safe trajectories are found based on "safety tubes" from model mismatches, integrating these into a discrepancy-aware cost map for optimal trajectory generation under safety constraints, even with an inaccurate reduced-order model, as detailed in 4.

Future Directions

An immediate extension of this research involves the integration of the moving obstacle avoidance prediction framework with the uncertainty-aware controller augmentation. This integration is poised to yield a robust model-predictive planning algorithm, particularly adept at navigating dynamic scenarios. A key facet of this development is the enhancement of obstacle detection and tracking capabilities, leveraging sensors such as LIDAR and cameras. My ongoing efforts are focused on incorporating computer vision techniques for refined obstacle identification and isolation, coupled with point cloud tracing for precise obstacle position extraction. This advancement is set to significantly bolster the existing safety-critical autonomy framework by facilitating the assimilation of real sensor inputs.

Looking forward, future endeavors are directed towards refining the offline conformal-driven discrepancy upper bound component into a more dynamic online algorithm. Inspired by the methodologies presented in [14], adaptive conformal prediction techniques can be modified to actively identify distribution shifts, which may arise from changes in terrain or environmental conditions. Additionally, there are plans to implement the proposed planner on GPU architectures and to transition from reduced-order models to more sophisticated learning-based models. However, further theoretical analysis is required to ascertain the discrepancy-aware model predictive path integral (MPPI) optimality and its recursive feasibility.

Another promising avenue is to incorporate higher-fidelity learning-based approaches, akin to those outlined in [92]. This shift from using simplified reduced-order models as detailed in 3 and 4 is expected to significantly improve the system's tracking performance. This is particularly relevant for ground vehicles operating in rough terrain, where high-fidelity models can account for variables such as friction, wheel-ground contact, and slope dependencies. Employing learning-based methods that predict future states from data sources like point clouds, RBGD images, and other state and input data is also advantageous. More precise models result in less conservative maximum tracking tubes derived from uncertainty quantification. Notably, the sampling-based planning method, MPPI, is highly compatible with learning-based models, as it does not require the system dynamics to be convex or continuous.

BIBLIOGRAPHY

[1] Fan, David D., Otsu, Kyohei, Kubo, Yuki, Dixit, Anushri, Burdick, Joel W., and Agha-Mohammadi, Ali-Akbar, “Step: Stochastic traversability evaluation and planning for risk-aware off-road navigation,” in *Robot.: Sci. Syst.*, 2021.

[2] Nakka, Yashwanth Kumar and Chung, Soon-Jo, “Trajectory optimization of chance-constrained nonlinear stochastic systems for motion planning under uncertainty,” *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 203–222, 2022.

[3] Blackmore, Lars, Ono, Masahiro, Bektassov, Askar, and Williams, Brian C., “A probabilistic particle-control approximation of chance-constrained stochastic predictive control,” *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 502–517, 2010. DOI: 10.1109/TRO.2010.2044948.

[4] Du Toit, Noel E. and Burdick, Joel W., “Robot motion planning in dynamic, uncertain environments,” *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 101–115, 2011.

[5] Bonadies, Stephanie, Lefcourt, Alan, and Gadsden, S. Andrew, “A survey of unmanned ground vehicles with applications to agricultural and environmental sensing,” in *Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping*, SPIE, vol. 9866, 2016, pp. 142–155.

[6] Ebadi, Kamak, Chang, Yun, Palieri, Matteo, Stephens, Alex, Hatteland, Alex, Heiden, Eric, Thakur, Abhishek, Funabiki, Nobuhiro, Morrell, Benjamin, Wood, Sally, Carlone, Luca, and Agha-Mohammadi, Ali-akbar, “Lamp: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 80–86. DOI: 10.1109/ICRA40945.2020.9197082.

[7] Wu, Yuzhan, Ding, Yuanhao, Ding, Susheng, Savaria, Yvon, and Li, Meng, “Autonomous last-mile delivery based on the cooperation of multiple heterogeneous unmanned ground vehicles,” *Mathematical Problems in Engineering*, vol. 2021, pp. 1–15, 2021.

[8] Thangavelautham, Jekan, Chandra, Aman, and Jensen, Erik, “Autonomous robot teams for lunar mining base construction and operation,” in *2020 IEEE Aerospace Conference*, 2020, pp. 1–16. DOI: 10.1109/AERO47225.2020.9172811.

[9] Dixit, Anushri, Fan, David D., Otsu, Kyohei, Dey, Sharmita, Agha-Mohammadi, Ali-Akbar, and Burdick, Joel W., “Step: Stochastic traversability evaluation

and planning for risk-aware off-road navigation; results from the darpa subterranean challenge," *arXiv preprint arXiv:2303.01614*, 2023.

[10] Bouman, Amanda, Ginting, Muhammad Fadhil, Alatur, Nikhilesh, Palieri, Matteo, Fan, David D., Touma, Thomas, Pailevanian, Torkom, Kim, Sung-Kyun, Otsu, Kyohei, Burdick, Joel W., and Agha-mohammadi, Ali-akbar, *Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion*, 2020. arXiv: `2010.09259 [cs.RO]`.

[11] Akella, Prithvi, Dixit, Anushri, Ahmadi, Mohamadreza, Burdick, Joel W., and Ames, Aaron D., "Sample-based bounds for coherent risk measures: Applications to policy synthesis and verification," *arXiv:2204.09833*, 2022.

[12] Saveriano, Matteo, Yin, Yuchao, Falco, Pietro, and Lee, Dongheui, "Data-efficient control policy search using residual dynamics learning," in *2017 Int. Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4709–4715.

[13] Akella, Prithvi, Wei, Skylar X., Burdick, Joel W., and Ames, Aaron, "Learning disturbances online for risk-aware control: Risk-aware flight with less than one minute of data," in *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, ser. Proceedings of Machine Learning Research, vol. 211, PMLR, 15–16 Jun 2023, pp. 665–678,

[14] Dixit, Anushri*, Lindemann, Lars*, Wei, Skylar X., Cleaveland, Matthew, Pappas, George J., and Burdick, Joel W., "Adaptive conformal prediction for motion planning among dynamic agents," in *Learning for Dynamics and Control Conference*, PMLR, 2023, pp. 300–314,

[15] Wei, Skylar X.*, Dixit, Anushri*, Tomar, Shashank, and Burdick, Joel W., "Moving obstacle avoidance: A data-driven risk-aware approach," *IEEE Control Systems Letters*, vol. 7, pp. 289–294, 2023. DOI: `10.1109/LCSYS.2022.3181191`,

[16] Wei, Skylar X., Dixit, Anushri, Brunton, Steven L., and Burdick, Joel W., "Data-driven moving obstacle avoidance: A planning framework with probabilistic safety guarantees," in *In preparation for The International Journal of Robotics Research*, Sage, 2024,

[17] Wei, Skylar X., Gan, Lu, and Burdick, Joel W., "A safety-critical framework for ugvs in complex environment: A data-driven discrepancy-aware approach," in *Preprint submitted to Robotics and Autonomous Systems*, Elsevier, 2024,

[18] Alan, Anil, Taylor, Andrew J., He, Chaozhe R., Orosz, Gábor, and Ames, Aaron D., "Safe controller synthesis with tunable input-to-state safe control barrier functions," *IEEE Control Systems Letters*, vol. 6, pp. 908–913, 2021.

[19] Csomay-Shanklin, Noel, Taylor, Andrew J., Rosolia, Ugo, and Ames, Aaron D., "Multi-rate planning and control of uncertain nonlinear systems: Model predictive control and control lyapunov functions," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, IEEE, 2022, pp. 3732–3739.

[20] Majumdar, Anirudha and Pavone, Marco, "How should a robot assess risk? towards an axiomatic theory of risk in robotics," in *Robotics Research*, Springer, 2020, pp. 75–84.

[21] Dixit, Anushri, Ahmadi, Mohamadreza, and Burdick, Joel W., "Risk-sensitive motion planning using entropic value-at-risk," in *2021 European Control Conference (ECC)*, IEEE, 2021, pp. 1726–1732.

[22] Cakmak, Sait, Astudillo Marban, Raul, Frazier, Peter, and Zhou, Enlu, "Bayesian optimization of risk measures," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 130–20 141, 2020.

[23] Makarova, Anastasia, Usmanova, Ilnura, Bogunovic, Ilija, and Krause, Andreas, "Risk-averse heteroscedastic bayesian optimization," *Advances in Neural Information Processing Systems*, vol. 34, pp. 17 235–17 245, 2021.

[24] O'Connell, Michael, Shi, Guanya, Shi, Xichen, Azizzadenesheli, Kamyar, Anandkumar, Anima, Yue, Yisong, and Chung, Soon-Jo, "Neural-fly enables rapid learning for agile flight in strong winds," *Science Robotics*, vol. 7, no. 66, 2022.

[25] Bemporad, Alberto and Morari, Manfred, "Robust model predictive control: A survey," in *Robustness in identification and control*, Springer, 2007, pp. 207–226.

[26] Lopez, Brett T., Slotine, Jean-Jacques E., and How, Jonathan P., "Dynamic tube mpc for nonlinear systems," in *2019 American Control Conference (ACC)*, IEEE, 2019, pp. 1655–1662.

[27] Zhu, Hai and Alonso-Mora, Javier, "Chance-constrained collision avoidance for mavs in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 776–783, 2019. DOI: `10.1109/LRA.2019.2893494`.

[28] Blackmore, Lars, Ono, Masahiro, and Williams, Brian C., "Chance-constrained optimal path planning with obstacles," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1080–1094, 2011. DOI: `10.1109/TRO.2011.2161160`.

[29] Calafiore, Giuseppe C. and Fagiano, Lorenzo, "Stochastic model predictive control of lpv systems via scenario optimization," *Automatica*, vol. 49, no. 6, pp. 1861–1866, 2013.

[30] Williams, Grady, Wagener, Nolan, Goldfain, Brian, Drews, Paul, Rehg, James M., Boots, Byron, and Theodorou, Evangelos A., "Information theoretic mpc for model-based reinforcement learning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1714–1721. DOI: `10.1109/ICRA.2017.7989202`.

[31] Williams, Grady, Goldfain, Brian, Drews, Paul, Saigol, Kamil, Rehg, James M., and Theodorou, Evangelos A., “Robust sampling based model predictive control with sparse objective information.,” in *Robotics: Science and Systems*, vol. 14, 2018, p. 2018.

[32] Gandhi, Manan S., Vlahov, Bogdan, Gibson, Jason, Williams, Grady, and Theodorou, Evangelos A., “Robust model predictive path integral control: Analysis and performance guarantees,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1423–1430, 2021. DOI: 10.1109/LRA.2021.3057563.

[33] Thrun, Sebastian, “Probabilistic robotics,” *Commun. ACM*, vol. 45, no. 3, pp. 52–57, Mar. 2002, ISSN: 0001-0782. DOI: 10.1145/504729.504754. [Online]. Available: https://doi.org/10.1145/504729.504754.

[34] Thrun, Sebastian and Bücken, Arno, “Integrating grid-based and topological maps for mobile robot navigation,” in *Proceedings of the national conference on artificial intelligence*, 1996, pp. 944–951.

[35] Tsardoulias, Emmanouil G., Iliakopoulou, A., Kargakos, Andreas, and Petrou, Loukas, “A review of global path planning methods for occupancy grid maps regardless of obstacle density,” *Journal of Intelligent & Robotic Systems*, vol. 84, pp. 829–858, 2016.

[36] Oleynikova, Helen, Millane, Alexander, Taylor, Zachary, Galceran, Enric, Nieto, Juan I., and Siegwart, Roland Y., “Signed distance fields: A natural representation for both mapping and planning,” 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:28083959.

[37] Camps, Gadiel S., Dyro, Robert, Pavone, Marco, and Schwager, Mac, *Learning deep sdf maps online for robot navigation and exploration*, 2022. arXiv: 2207.10782 [cs.RO].

[38] Fan, David D., Agha-mohammadi, Ali-akbar, and Theodorou, Evangelos A., “Learning risk-aware costmaps for traversability in challenging environments,” *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 279–286, 2022. DOI: 10.1109/LRA.2021.3125047.

[39] Cai, Xiaoyi, Ancha, Siddharth, Sharma, Lakshay, Osteen, Philip R., Bucher, Bernadette, Phillips, Stephen, Wang, Jiuguang, Everett, Michael, Roy, Nicholas, and How, Jonathan P., *Evora: Deep evidential traversability learning for risk-aware off-road autonomy*, 2023. arXiv: 2311.06234 [cs.RO].

[40] Sharma, Lakshay, Everett, Michael, Lee, Donggun, Cai, Xiaoyi, Osteen, Philip, and How, Jonathan, “Ramp: A risk-aware mapping and planning pipeline for fast off-road ground robot navigation,” Sep. 2022. DOI: 10.48550/arXiv.2210.06605.

[41] Yin, Ji, Dawson, Charles, Fan, Chuchu, and Tsiotras, Panagiotis, *Shield model predictive path integral: A computationally efficient robust mpc approach using control barrier functions*, 2023. arXiv: 2302.11719 [cs.RO].

[42] Clark Jack and Amodei, Dario, *Faulty reward functions in the wild*. [Online]. Available: `https://openai.com/research/faulty-reward-functions`.

[43] Gan, Lu, Grizzle, Jessy W., Eustice, Ryan M., and Ghaffari, Maani, "Energy-based legged robots terrain traversability modeling via deep inverse reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8807–8814, 2022. DOI: `10.1109/LRA.2022.3188100`.

[44] Frey, Jonas, Mattamala, Matias, Chebrolu, Nived, Cadena, Cesar, Fallon, Maurice, and Hutter, Marco, "Fast traversability estimation for wild visual navigation," *arXiv preprint arXiv:2305.08510*, 2023.

[45] Kamil, Farah, Tang, S., Khaksar, Weria, Zulkifli, N., and Ahmad, Siti Azfanizam, "A review on motion planning and obstacle avoidance approaches in dynamic environments," *Advances in Robotics & Automation*, vol. 4, no. 2, pp. 134–142, 2015.

[46] Pandey, Anish, Pandey, Shalini, and Parhi, DR, "Mobile robot navigation and obstacle avoidance techniques: A review," *Int Rob Auto J*, vol. 2, no. 3, p. 00 022, 2017.

[47] Lam, C.P., Chouand, C.T. Chiang, K.H., and Fu, L.C., "Human-centered robot navigation—towards a harmoniously human–robot coexisting environment," *IEEE Trans. Robotics*, vol. 27, no. 1, pp. 99–112, 2011.

[48] van den Berg, J., Ferguson, D., and Kuffner, J., "Anytime path planning and replanning in dynamic environments," in *IEEE ICRA*, 2006. DOI: `10.1109/ROBOT.2006.1642056`.

[49] Ames, Aaron D., Grizzle, Jessy W., and Tabuada, Paulo, "Control barrier function based quadratic programs with application to adaptive cruise control," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 6271–6278. DOI: `10.1109/CDC.2014.7040372`.

[50] Borrmann, Urs, Wang, Li, Ames, Aaron D., and Egerstedt, Magnus, "Control barrier certificates for safe swarm behavior," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 68–73, 2015, Analysis and Design of Hybrid Systems ADHS, ISSN: 2405-8963. DOI: `https://doi.org/10.1016/j.ifacol.2015.11.154`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S240589631502412X`.

[51] Cheng, Richard, Khojasteh, Mohammad Javad, Ames, Aaron D., and Burdick, Joel W., "Safe multi-agent interaction through robust control barrier functions with learned uncertainties," in *2020 59th IEEE Conference on Decision and Control (CDC)*, IEEE, 2020, pp. 777–783.

[52] Wang, Li, Ames, Aaron, and Egerstedt, Magnus, "Safety barrier certificates for heterogeneous multi-robot systems," in *2016 American control conference (ACC)*, IEEE, 2016, pp. 5213–5218.

[53] Singletary, Andrew, Nilsson, Petter, Gurriet, Thomas, and Ames, Aaron D., "Online active safety for robotic manipulators," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 173–178.

[54] P., Varun V., Vinod, Abraham P., and Kolathaya, Shishir, "Motion planning with dynamic obstacles using convexified control barrier functions," in *2021 Seventh Indian Control Conference (ICC)*, 2021, pp. 81–86. DOI: `10.1109/ICC54714.2021.9703149`.

[55] Chen, Yuxiao, Singletary, Andrew, and Ames, Aaron D., "Guaranteed obstacle avoidance for multi-robot operations with limited actuation: A control barrier function approach," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 127–132, 2021. DOI: `10.1109/LCSYS.2020.3000748`.

[56] Haraldsen, Aurora, Wiig, Martin S., Ames, Aaron D., and Pettersen, Kristin Y., "Safety-critical control of nonholonomic vehicles in dynamic environments using velocity obstacles," *arXiv preprint arXiv:2310.00713*, 2023.

[57] Tordesillas, Jesus and How, Jonathan P., "Panther: Perception-aware trajectory planner in dynamic environments," *IEEE Access*, vol. 10, pp. 22 662–22 677, 2022.

[58] Kondo, Kota, Figueroa, Reinaldo, Rached, Juan, Tordesillas, Jesus, Lusk, Parker C., and How, Jonathan P., *Robust mader: Decentralized multiagent trajectory planner robust to communication delay in dynamic environments*, 2023. arXiv: `2303.06222 [cs.RO]`.

[59] Hyndman, Rob J. and Athanasopoulos, George, *Forecasting: principles and practice*. OTexts, 2018.

[60] Schwarm, Alexander T. and Nikolaou, Michael, "Chance-constrained model predictive control," *AIChE Journal*, vol. 45, no. 8, pp. 1743–1752, 1999.

[61] Zhu, Hai and Alonso-Mora, Javier, "Chance-constrained collision avoidance for mavs in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 776–783, 2019.

[62] Castillo-Lopez, Manuel, Ludivig, Philippe, Sajadi-Alamdari, Seyed Amin, Sanchez-Lopez, Jose Luis, Olivares-Mendez, Miguel A., and Voos, Holger, "A real-time approach for chance-constrained motion planning with dynamic obstacles," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3620–3625, 2020.

[63] Zhang, Xiaoxue, Ma, Jun, Cheng, Zilong, Huang, Sunan, Ge, Shuzhi Sam, and Lee, Tong Heng, "Trajectory generation by chance-constrained nonlinear mpc with probabilistic prediction," *IEEE Transactions on Cybernetics*, vol. 51, no. 7, pp. 3616–3629, 2021. DOI: `10.1109/TCYB.2020.3032711`.

[64] Lin, Jiahao, Zhu, Hai, and Alonso-Mora, Javier, "Robust vision-based obstacle avoidance for micro aerial vehicles in dynamic environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 2682–2688.

[65] Brunton, Steven, Brunton, Bingni, Proctor, Joshua L., Kaiser, Eurika, and Kutz, J. Nathan, "Chaos as an intermittently forced linear system," *Nat. commun.*, 2017.

[66] Takens, Floris, "Detecting strange attractors in turbulence," in *Dynamical systems and turbulence*, Springer, 1981, pp. 366–381.

[67] Kamb, Mason, Kaiser, Eurika, Brunton, Steven L., and Kutz, J. Nathan, "Time-delay observables for koopman: Theory and applications," *SIAM Journal on Applied Dynamical Systems*, vol. 19, no. 2, pp. 886–917, 2020. DOI: 10.1137/18M1216572.

[68] Hirsh, Seth M., Ichinaga, Sara M., Brunton, Steven L., Nathan Kutz, J., and Brunton, Bingni W., "Structured time-delay models for dynamical systems with connections to frenet–serret frame," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 477, no. 2254, p. 20 210 097, 2021. DOI: 10.1098/rspa.2021.0097.

[69] Pan, Shaowu and Duraisamy, Karthik, "On the structure of time-delay embedding in linear models of non-linear dynamical systems," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 30, no. 7, p. 073 135, Jul. 2020, ISSN: 1054-1500. DOI: 10.1063/5.0010886. eprint: https://pubs.aip.org/aip/cha/article-pdf/doi/10.1063/5.0010886/14631039/073135_1_online.pdf. [Online]. Available: https://doi.org/10.1063/5.0010886.

[70] Farid, Alec, Veer, Sushant, Ivanovic, Boris, Leung, Karen, and Pavone, Marco, "Task-relevant failure detection for trajectory predictors in autonomous vehicles," *arXiv preprint arXiv:2207.12380*, 2022.

[71] Luo, Rachel, Zhao, Shengjia, Kuck, Jonathan, Ivanovic, Boris, Savarese, Silvio, Schmerling, Edward, and Pavone, Marco, "Sample-efficient safety assurances using conformal prediction," *arXiv preprint arXiv:2109.14082*, 2021.

[72] Yi, Shuai, Li, Hongsheng, and Wang, Xiaogang, "Pedestrian behavior understanding and prediction with deep neural networks," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, Springer, 2016, pp. 263–279.

[73] Ridel, Daniela, Rehder, Eike, Lauer, Martin, Stiller, Christoph, and Wolf, Denis, "A literature review on the prediction of pedestrian behavior in urban scenarios," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 3105–3112. DOI: 10.1109/ITSC.2018.8569415.

[74] Fisac, Jaime F., Bajcsy, Andrea, Herbert, Sylvia L., Fridovich-Keil, David, Wang, Steven, Tomlin, Claire J., and Dragan, Anca D., "Probabilistically safe robot planning with confidence-based human predictions," in *14th Robotics: Science and Systems, RSS 2018*, MIT Press Journals, 2018.

[75] Nakamura, Kensuke and Bansal, Somil, "Online update of safety assurances using confidence-based predictions," *arXiv preprint arXiv:2210.01199*, 2022.

[76] Fridovich-Keil, David, Bajcsy, Andrea, Fisac, Jaime F., Herbert, Sylvia L., Wang, Steven, Dragan, Anca D., and Tomlin, Claire J., "Confidence-aware motion prediction for real-time collision avoidance1," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 250–265, 2020.

[77] Bansal, Somil, Bajcsy, Andrea, Ratner, Ellis, Dragan, Anca D., and Tomlin, Claire J., "A hamilton-jacobi reachability-based framework for predicting and analyzing human motion for safe planning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 7149–7155.

[78] Liu, Jinsun, Adu, Challen E., Lymburner, Lucas, Kaushik, Vishrut, Trang, Lena, and Vasudevan, Ram, *Radius: Risk-aware, real-time, reachability-based motion planning*, 2023. arXiv: `2302.07933 [cs.RO]`.

[79] Holmes, Patrick, Kousik, Shreyas, Zhang, Bohao, Raz, Daphna, Barbalata, Corina, Johnson-Roberson, Matthew, and Vasudevan, Ram, "Reachable sets for safe, real-time manipulator trajectory design," *arXiv*, vol. arXiv:2002.01591, 2020.

[80] Lindqvist, Bjorn, Mansouri, Sina S., Agha-mohammadi, Ali-akbar, and Nikolakopoulos,George, "Nonlinear mpc for collision avoidance and control of uavs with dynamic obstacles," *IEEE Robot. & Autom. Lett*, 2020.

[81] Hwang, Inseok and Seah, Chze Eng, "Intent-based probabilistic conflict detection for the next generation air transportation system," *Proceedings of the IEEE*, vol. 96, no. 12, pp. 2040–2059, 2008. DOI: `10.1109/JPROC.2008.2006138`.

[82] Busch, Finn Lukas, Johnson, Jake, Zhu, Edward L., and Borrelli, Francesco, *A gaussian process model for opponent prediction in autonomous racing*, 2022. DOI: `10.48550/ARXIV.2204.12533`. [Online]. Available: `https://arxiv.org/abs/2204.12533`.

[83] Choi, Sungjoon, Kim, Eunwoo, Lee, Kyungjae, and Oh, Songhwai, "Real-time nonparametric reactive navigation of mobile robots in dynamic environments," *Robotics and Autonomous Systems*, vol. 91, pp. 11–24, 2017.

[84] Omainska, Marco, Yamauchi, Junya, Beckers, Thomas, Hatanaka, Takeshi, Hirche, Sandra, and Fujita, Masayuki, "Gaussian process-based visual pursuit control with unknown target motion learning in three dimensions," *SICE Journal of Control, Measurement, and System Integration*, vol. 14, no. 1, pp. 116–127, 2021.

[85] Fulgenzi, Chiara, Tay, Christopher, Spalanzani, Anne, and Laugier, Christian, "Probabilistic navigation in dynamic environment using rapidly-exploring random trees and gaussian processes," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2008, pp. 1056–1062.

[86] Lindemann, Lars, Cleaveland, Matthew, Shim, Gihyun, and Pappas, George J., "Safe planning in dynamic environments using conformal prediction," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 5116–5123, 2023. DOI: 10.1109/LRA.2023.3292071.

[87] Golyandina, Nina, Nekrutkin, Vladimir, and Zhigljavsky, Anatoly A., *Analysis of time series structure: SSA and related techniques*. CRC press, 2001.

[88] Golyandina, Nina and Korobeynikov, A., "Basic singular spectrum analysis and forecasting with r," *Comp. Stat. & Data Anal.*,

[89] Fasel, Urban, Kutz, J. Nathan, Brunton, Bingni, and Brunton, Steven, "Ensemble-sindy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control," *Proceedings of Royal Society A*, vol. 478, no. 2260, 2022.

[90] Morgan, Daniel, Chung, Soon-Jo, and Hadaegh, Fred, "Model predictive control of swarms of spacecraft using sequential convex programming," *Journal of Guidance, Control, and Dynamics*, vol. 37, pp. 1–16, Apr. 2014. DOI: 10.2514/1.G000218.

[91] Augugliaro, Federico, Schoellig, Angela P., and D'Andrea, Raffaello, "Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach," *International Conference on Intelligent Robots System*, 2012.

[92] Folkestad, Carl, Wei, Skylar X., and Burdick, Joel W., "Koopnet: Joint learning of koopman bilinear models and function dictionaries with application to quadrotor trajectory tracking," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 1344–1350,

[93] Koopman, Bernard O. and Neumann, John Von, "Dynamical systems of continuous spectra," *Proceedings of the National Academy of Sciences*, vol. 18, no. 3, pp. 255–263, 1932.

[94] Koopman, Bernard O., "Hamiltonian systems and transformation in hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931.

[95] Mezić, Igor, "Spectral properties of dynamical systems, model reduction and decompositions," *Nonlinear Dynamics*, vol. 41, pp. 309–325, 2005.

[96] Taira, Kunihiko, Brunton, Steven L., Dawson, Scott TM, Rowley, Clarence W., Colonius, Tim, McKeon, Beverley J., Schmidt, Oliver T., Gordeyev, Stanislav, Theofilis, Vassilios, and Ukeiley, Lawrence S., "Modal analysis

of fluid flows: An overview," *AIAA Journal*, vol. 55, no. 12, pp. 4013–4041, 2017.

[97] Brunton, Bingni W., Johnson, Lise A., Ojemann, Jeffrey G., and Kutz, J. Nathan, "Extracting spatial–temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition," *Journal of Neuroscience Methods*, vol. 258, pp. 1–15, 2016, ISSN: 0165-0270. DOI: `https://doi.org/10.1016/j.jneumeth.2015.10.010`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0165027015003829`.

[98] Mauroy, Alexandre, Susuki, Y., and Mezić, Igor, *Koopman operator in systems and control*. Springer, 2020.

[99] Mezić, Igor, "Spectrum of the koopman operator, spectral expansions in functional spaces, and state-space geometry," *Journal of Nonlinear Science*, vol. 30, no. 5, pp. 2091–2145, 2020.

[100] Arbabi, Hassan and Mezic, Igor, "Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator," *SIAM Journal on Applied Dynamical Systems*, vol. 16, no. 4, pp. 2096–2126, 2017.

[101] Kaiser, Eurika, Kutz, J. Nathan, and Brunton, Steven L., "Data-driven approximations of dynamical systems operators for control," *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications*, pp. 197–234, 2020.

[102] Coulson, Jeremy, Lygeros, John, and Dörfler, Florian, "Distributionally robust chance constrained data-enabled predictive control," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3289–3304, 2022. DOI: `10.1109/TAC.2021.3097706`.

[103] Fazel, Maryam, Pong, Ting Kei, Sun, Defeng, and Tseng, Paul, "Hankel matrix rank minimization with applications to system identification and realization," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 3, pp. 946–977, 2013.

[104] Vovk, Vladimir, Gammerman, Alex, and Shafer, Glenn, *Algorithmic Learning in a Random World*. Berlin, Heidelberg: Springer-Verlag, 2005, ISBN: 0387001522.

[105] Angelopoulos, Anastasios N. and Bates, Stephen, "A gentle introduction to conformal prediction and distribution-free uncertainty quantification," *arXiv preprint arXiv:2107.07511*, 2021.

[106] Chow, Yuan Shih and Teicher, Henry, *Probability theory: independence, interchangeability, martingales*. Springer Science & Business Media, 2003.

[107] Tibshirani, Ryan J., Foygel Barber, Rina, Candes, Emmanuel, and Ramdas, Aaditya, "Conformal prediction under covariate shift," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[108] Sadinle, Mauricio, Lei, Jing, and Wasserman, Larry, "Least ambiguous set-valued classifiers with bounded error levels," *Journal of the American Statistical Association*, vol. 114, no. 525, pp. 223–234, 2019.

[109] Gibbs, Isaac and Candes, Emmanuel, "Adaptive conformal inference under distribution shift," *Advances in Neural Information Processing Systems*, vol. 34, pp. 1660–1672, 2021.

[110] Zaffran, Margaux, Féron, Olivier, Goude, Yannig, Josse, Julie, and Dieuleveut, Aymeric, "Adaptive conformal predictions for time series," in *International Conference on Machine Learning*, PMLR, 2022, pp. 25 834–25 866.

[111] Gibbs, Isaac and Candès, Emmanuel, "Conformal inference for online prediction with arbitrary distribution shifts," *arXiv preprint arXiv:2208.08401*, 2022.

[112] Bastani, Osbert, Gupta, Varun, Jung, Christopher, Noarov, Georgy, Ramalingam, Ramya, and Roth, Aaron, "Practical adversarial multivalid conformal prediction," *arXiv preprint arXiv:2206.01067*, 2022.

[113] Williams, Grady, Drews, Paul, Goldfain, Brian, Rehg, James M., and Theodorou, Evangelos A., "Information-theoretic model predictive control: Theory and applications to autonomous driving," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1603–1622, 2018. DOI: 10.1109/TRO.2018.2865891.

[114] Williams, Grady, Drews, Paul, Goldfain, Brian, Rehg, James M., and Theodorou, Evangelos A., "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1433–1440. DOI: 10.1109/ICRA.2016.7487277.

[115] Gros, Sébastien, Zanon, Mario, Quirynen, Rien, Bemporad, Alberto, and Diehl, Moritz, "From Linear to Nonlinear MPC: Bridging the Gap via the Real-time Iteration," *International Journal of Control*, vol. 93, no. 1, pp. 62–80, 2020, ISSN: 13665820. DOI: 10.1080/00207179.2016.1222553.

[116] Rawlings, James B. and Mayne, David Q., *Model Predictive Control Theory and Design*. Nob Hill Publishing, 2012, ISBN: 978-0-975-93770-9.

[117] Wei, Skylar X., Harderup, Peder, and Burdick, Joel W., *Differential flatness and flatness inspired control of aerial manipulators based on lagrangian reduction*, 2021. arXiv: 2111.01302 [cs.RO],

[118] Diehl, Moritz and Bjornberg, Jakob, "Robust dynamic programming for min-max model predictive control of constrained uncertain systems," *IEEE Transactions on Automatic Control*, vol. 49, no. 12, pp. 2253–2257, 2004.

[119] Ubellacker, Wyatt, Csomay-Shanklin, Noel, Molnar, Tamas G, and Ames, Aaron D, “Verifying safe transitions between dynamic motion primitives on legged robots,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021, pp. 8477–8484.

[120] Xiong, Xiaobin, “Reduced order model inspired robotic bipedal walking: A step-to-step dynamics approximation based approach,” Ph.D. dissertation, California Institute of Technology, 2021.

[121] Thieffry, Maxime, Kruszewski, Alexandre, Duriez, Christian, and Guerra, Thierry-Marie, “Control design for soft robots based on reduced-order model,” *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 25–32, 2018.

[122] Kim, Youngmin, Allmendinger, Richard, and López-Ibáñez, Manuel, “Safe learning and optimization techniques: Towards a survey of the state of the art,” in *International Workshop on the Foundations of Trustworthy AI Integrating Learning, Optimization and Reasoning*, Springer, 2020, pp. 123–139.

[123] Kolathaya, Shishir and Ames, Aaron D., “Input-to-state safety with control barrier functions,” *IEEE control systems letters*, vol. 3, no. 1, pp. 108–113, 2018.

[124] Ahmadi, Mohamadreza, Xiong, Xiaobin, and Ames, Aaron D., “Risk-sensitive path planning via cvar barrier functions: Application to bipedal locomotion,” *arXiv preprint arXiv:2011.01578*, 2020.

[125] Akella, Prithvi, Ubellacker, Wyatt, and Ames, Aaron D., “Safety-critical controller verification via sim2real gap quantification,” *arXiv preprint arXiv: 2209.09337*, 2022.

[126] Romdlony, Muhammad Zakiyullah and Jayawardhana, Bayu, “On the new notion of input-to-state safety,” in *2016 IEEE 55th conference on decision and control (CDC)*, IEEE, 2016, pp. 6403–6409.

[127] Taylor, Andrew, Singletary, Andrew, Yue, Yisong, and Ames, Aaron, “Learning for safety-critical control with control barrier functions,” in *Learning for Dynamics and Control*, PMLR, 2020, pp. 708–717.

[128] Lindemann, Lars, Pappas, George J., and Dimarogonas, Dimos V., “Reactive and risk-aware control for signal temporal logic,” *IEEE Transactions on Automatic Control*, 2021.

[129] Buisson-Fenet, Mona, Solowjow, Friedrich, and Trimpe, Sebastian, “Actively learning gaussian process dynamics,” in *Learning for Dynamics and Control*, PMLR, 2020, pp. 5–15.

[130] Nguyen-Tuong, Duy and Peters, Jan, “Model learning for robot control: A survey,” *Cognitive processing*, vol. 12, no. 4, pp. 319–340, 2011.

[131] Jain, Achin, Nghiem, Truong, Morari, Manfred, and Mangharam, Rahul, “Learning and control using gaussian processes,” in *2018 ACM/IEEE 9th international conference on cyber-physical systems (ICCPS)*, IEEE, 2018, pp. 140–149.

[132] Berkenkamp, Felix and Schoellig, Angela P., “Safe and robust learning control with gaussian processes,” in *2015 European Control Conference (ECC)*, IEEE, 2015, pp. 2496–2501.

[133] Westenbroek, Tyler, Agrawal, Ayush, Castañeda, Fernando, Sastry, S Shankar, and Sreenath, Koushil, “Combining model-based design and model-free policy optimization to learn safe, stabilizing controllers,” *IFAC-PapersOnLine*, vol. 54, no. 5, pp. 19–24, 2021.

[134] Wang, Li, Theodorou, Evangelos A, and Egerstedt, Magnus, “Safe learning of quadrotor dynamics using barrier certificates,” in *2018 Int. Conference on Robotics and Automation (ICRA)*, 2018, pp. 2460–2465.

[135] Williams, Christopher K. and Rasmussen, Carl E., *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2.

[136] Ahmadi, Mohamadreza, Rosolia, Ugo, Ingham, Michel D., Murray, Richard M., and Ames, Aaron D., “Constrained risk-averse markov decision processes,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 11 718–11 725.

[137] Ono, Masahiro, Heverly, Matthew, Rothrock, Brandon, Almeida, Eduardo, Calef, Fred, Soliman, Tariq, Williams, Nathan, Gengl, Hallie, Ishimatsu, Takuto, Nicholas, Austin, *et al.*, “Mars 2020 site-specific mission performance analysis: Part 2. surface traversability,” in *2018 AIAA SPACE and Astronautics Forum and Exposition*, 2018, p. 5419.

[138] Heger, Matthias, “Consideration of risk in reinforcement learning,” in *Machine Learning Proceedings 1994*, Elsevier, 1994, pp. 105–111.

[139] Chow, Yinlam, Ghavamzadeh, Mohammad, Janson, Lucas, and Pavone, Marco, “Risk-constrained reinforcement learning with percentile risk criteria,” *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6070–6120, 2017.

[140] Mihatsch, Oliver and Neuneier, Ralph, “Risk-sensitive reinforcement learning,” *Machine learning*, vol. 49, no. 2, pp. 267–290, 2002.

[141] Geibel, Peter and Wysotzki, Fritz, “Risk-sensitive reinforcement learning applied to control under constraints,” *Journal of Artificial Intelligence Research*, vol. 24, pp. 81–108, 2005.

[142] Srinivas, Niranjan, Krause, Andreas, Kakade, Sham M., and Seeger, Matthias, “Gaussian process optimization in the bandit setting: No regret and experimental design,” *arXiv preprint arXiv:0912.3995*, 2009.

[143] Chowdhury, Sayak Ray and Gopalan, Aditya, "On kernelized multi-armed bandits," in *International Conference on Machine Learning*, PMLR, 2017, pp. 844–853.

[144] Linsmeier, Thomas J. and Pearson, Neil D., "Value at risk," *Financial Analysts Journal*, vol. 56, no. 2, pp. 47–67, 2000.

[145] Johannink, Tobias, Bahl, Shikhar, Nair, Ashvin, Luo, Jianlan, Kumar, Avinash, Loskyll, Matthias, Ojea, Juan Aparicio, Solowjow, Eugen, and Levine, Sergey, "Residual reinforcement learning for robot control," in *2019 Int. Conference on Robotics and Automation (ICRA)*, 2019, pp. 6023–6029.

[146] Schperberg, Alexander, Tanaka, Yusuke, Xu, Feng, Menner, Marcel, and Hong, Dennis, "Real-to-sim: Deep learning with auto-tuning to predict residual errors using sparse data," *arXiv preprint arXiv:2209.03210*, 2022.

[147] Zeng, Andy, Song, Shuran, Lee, Johnny, Rodriguez, Alberto, and Funkhouser, Thomas, "Tossingbot: Learning to throw arbitrary objects with residual physics," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1307–1319, 2020.

[148] Meier, Lorenz, Honegger, Dominik, and Pollefeys, Marc, "Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6235–6240. DOI: `10.1109/ICRA.2015.7140074`.

[149] O'Connell, Michael, Shi, Guanya, Shi, Xichen, Azizzadenesheli, Kamyar, Anandkumar, Anima, Yue, Yisong, and Chung, Soon-Jo, "Neural-fly enables rapid learning for agile flight in strong winds," *Science Robotics*, vol. 7, no. 66, eabm6597, 2022. DOI: `10.1126/scirobotics.abm6597`.

[150] *Video*. [Online]. Available: `https://youtu.be/4i2GNU8ahSU`.

[151] Klancar, Gregor, Zdesar, Andrej, Blazic, Saso, and Skrjanc, Igor, *Wheeled mobile robotics: from fundamentals towards autonomous systems*. Butterworth-Heinemann, 2017.

[152] Aicardi, Michele, Casalino, Giuseppe, Bicchi, Antonio, and Balestrino, Aldo, "Closed loop steering of unicycle like vehicles via lyapunov techniques," *IEEE Robotics and Automation Magazine*, vol. 2, no. 1, pp. 27–35, 1995.

[153] De Luca, Alessandro, Oriolo, Giuseppe, and Vendittelli, Marilena, "Control of wheeled mobile robots: An experimental overview," in *Ramsete: Articulated and Mobile Robotics for Services and Technologies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 181–226.

[154] Khalil, Hassan K., *Nonlinear systems*. Prentice Hall, 2002.

[155] Chakraborty, Nilanjan and Ghosal, Ashitava, “Kinematics of wheeled mobile robots on uneven terrain,” *Mechanism and Machine Theory*, vol. 39, no. 12, pp. 1273–1287, 2004, 11th National Conference on Machines and Mechanisms (NaCoMM-2003), ISSN: 0094-114X. DOI: `https://doi.org/10.1016/j.mechmachtheory.2004.05.016`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0094114X04001478`.

[156] Sinha, Rohan, Harrison, James, Richards, Spencer M., and Pavone, Marco, “Adaptive robust model predictive control with matched and unmatched uncertainty,” in *2022 American Control Conference (ACC)*, IEEE, 2022, pp. 906–913.

[157] Cao, Wen-Jun and Xu, Jian-Xin, “Nonlinear integral-type sliding surface for both matched and unmatched uncertain systems,” *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1355–1360, 2004. DOI: `10.1109/TAC.2004.832658`.

[158] Pravitra, Jintasit, Ackerman, Kasey A., Cao, Chengyu, Hovakimyan, Naira, and Theodorou, Evangelos A., “L_1-adaptive mppi architecture for robust and agile control of multirotors,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 7661–7666.

[159] Ren, Allen Z., Dixit, Anushri, Bodrova, Alexandra, Singh, Sumeet, Tu, Stephen, Brown, Noah, Xu, Peng, Takayama, Leila, Xia, Fei, Varley, Jake, Xu, Zhenjia, Sadigh, Dorsa, Zeng, Andy, and Majumdar, Anirudha, *Robots that ask for help: Uncertainty alignment for large language model planners*, 2023. arXiv: `2307.01928 [cs.RO]`.

[160] Gronwall, Thomas H., “Note on the derivatives with respect to a parameter of the solutions of a system of differential equations,” *Annals of Mathematics*, vol. 20, no. 4, pp. 292–296, 1919, ISSN: 0003486X. (visited on 09/03/2023).

[161] Bastos Jr, Guaraci and Franco, Enrico, “Energy shaping dynamic tube-mpc for underactuated mechanical systems,” *Nonlinear Dynamics*, vol. 106, no. 1, pp. 359–380, 2021.

[162] Thrun, Sebastian, “Probabilistic robotics,” *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[163] *Supplementary video*. [Online]. Available: `https://youtu.be/0tZagFMfodI`.

[164] Hugemann, Wolfgang, “Driver reaction times in road traffic,” 2002. [Online]. Available: `https://api.semanticscholar.org/CorpusID:58933066`.

[165] Cosner, Ryan, Tucker, Maegan, Taylor, Andrew, Li, Kejun, Molnar, Tamas, Ubelacker, Wyatt, Alan, Anil, Orosz, Gabor, Yue, Yisong, and Ames, Aaron, “Safety-aware preference-based learning for safety-critical control,” in *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, R. Firoozi, N. Mehr, E. Yel, R. Antonova, J. Bohg, M. Schwager, and

M. Kochenderfer, Eds., ser. Proceedings of Machine Learning Research, vol. 168, PMLR, 23–24 Jun 2022, pp. 1020–1033.

[166] Tucker, Maegan, Li, Kejun, Yue, Yisong, and Ames, Aaron D., *Polar: Preference optimization and learning algorithms for robotics*, 2022. arXiv: `2208.04404 [cs.RO]`.

[167] Hummel, S.A., *Frisbee flight simulation and throw biomechanics*. University of California, Davis, 2003.

[168] Adbi, H. and Williams, L.J., "Principal component analysis," *Wiley interdisc. reviews: comp. statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[169] Golyandina, Nina, Nekrutkin, Vladimir, and Zhigljavsky, Anatoly, "Analysis of time series structure - ssa and related techniques," in *Monographs on statistics and applied probability*, 2001.

[170] Agarwal, A., Alomar, A., and Shah, D., "On multivariate singular spectrum analysis," *arXiv preprint arXiv:2006.13448*, Jun. 2020.

[171] Ghaoui, L., Oks, M., and Oustry, F., "Worst-case value-at-risk and robust portfolio optimization: A conic programming approach," *Operations Research*, vol. 51, pp. 543–556, Aug. 2003. DOI: `10.1287/opre.51.4.543.16101`.

[172] Foust, Rebecca, Chung, Soon-Jo, and Hadaegh, Fred Y., "Optimal guidance and control with nonlinear dynamics using sequential convex programming," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 4, pp. 633–644, 2020.

[173] Gurobi Optimization, LLC, *Gurobi Optimizer Ref. Manual*, 2022. [Online]. Available: `https://www.gurobi.com`.

[174] Lindqvist, Björn, Sopasakis, Pantelis, and Nikolakopoulos, George, "A scalable distributed collision avoidance scheme for multi-agent uav systems," in *2021 IROS*, IEEE, 2021, pp. 9212–9218.

[175] Lindemann, Lars, Cleaveland, Matthew, Shim, Gihyun, and Pappas, George J., "Safe planning in dynamic environments using conformal prediction," *arXiv preprint arXiv:2210.10254*, 2022.

[176] Shinbrot,Troy, Grebogi, Celso A., Wisdom, Jack, and Yorke, James A., "Chaos in a double pendulum," in *American Journal of Physics, American Association of Physics Teachers*, 1992.

[177] Dixit, Anushri, Lindemann, Lars, Wei, Skylar W., Cleaveland, Matthew, Pappas, George J., and Burdick, Joel W., "Adaptive conformal prediction for motion planning among dynamic agents (supplementary)," *arXiv preprint arXiv:2212.00278*, 2022.

[178] Hewing, Lukas, Wabersich, Kim P., and Zeilinger, Melanie N., "Recursively feasible stochastic model predictive control using indirect feedback," *Automatica*, vol. 119, p. 109 095, 2020, ISSN: 0005-1098. DOI: `https://doi.org/10.1016/j.automatica.2020.109095`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0005109820302934`.

[179] Singletary, Andrew, Swann, Aiden, Rodriguez, Ivan Dario Jimenez, and Ames, Aaron D., *Safe drone flight with time-varying backup controllers*, 2022. DOI: `10.48550/ARXIV.2207.05220`. [Online]. Available: `https://arxiv.org/abs/2207.05220`.

[180] Tordesillas, Jesus, Lopez, Brett T., Everett, Michael, and How, Jonathan P., *Faster: Fast and safe trajectory planner for navigation in unknown environments*, 2020. DOI: `10.48550/ARXIV.2001.04420`. [Online]. Available: `https://arxiv.org/abs/2001.04420`.

[181] Nair, Siddharth H., Lee, Hotae, Joa, Eunhyek, Wang, Yan, Tseng, H. Eric, and Borrelli, Francesco, *Predictive control for autonomous driving with uncertain, multi-modal predictions*, 2023. arXiv: `2310.20561 [cs.RO]`.

[182] Lindemann, Lars, Cleaveland, Matthew, Shim, Gihyun, and Pappas, George J., "Safe planning in dynamic environments using conformal prediction," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 5116–5123, 2023. DOI: `10.1109/LRA.2023.3292071`.

[183] Hanover, Drew, Loquercio, Antonio, Bauersfeld, Leonard, Romero, Angel, Penicka, Robert, Song, Yunlong, Cioffi, Giovanni, Kaufmann, Elia, and Scaramuzza, Davide, "Past, present, and future of autonomous drone racing: A survey," *arXiv preprint arXiv:2301.01755*, 2023.

[184] Tomlin, Claire, Pappas, George J., and Sastry, S. Shankar, "Conflict resolution for air traffic management: A study in multiagent hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 509–521, 1998. DOI: `10.1109/9.664154`.

[185] Castillo-Lopez, Manuel, Ludivig, Philippe, Sajadi-Alamdari, Seyed Amin, Sanchez-Lopez, Jose Luis, Olivares-Mendez, Miguel A., and Voos, Holger, "A real-time approach for chance-constrained motion planning with dynamic obstacles," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3620–3625, 2020. DOI: `10.1109/LRA.2020.2975759`.

[186] Kamel, Mina, Alonso-Mora, Javier, Siegwart, Roland, and Nieto, Juan, "Robust collision avoidance for multiple micro aerial vehicles using nonlinear model predictive control," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 236–243. DOI: `10.1109/IROS.2017.8202163`.

[187] Mezić, Igor, "On numerical approximations of the koopman operator," *Mathematics*, vol. 10, no. 7, 2022, ISSN: 2227-7390. DOI: 10.3390/math10071180.

[188] Brunton, Steven L., Brunton, Bingni W., Proctor, Joshua L., and Kutz, J. Nathan, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PLOS one*, vol. 11, no. 2, e0150171, 2016.

[189] Abdar, Moloud, Pourpanah, Farhad, Hussain, Sadiq, Rezazadegan, Dana, Liu, Li, Ghavamzadeh, Mohammad, Fieguth, Paul, Cao, Xiaochun, Khosravi, Abbas, Acharya, U. Rajendra, Makarenkov, Vladimir, and Nahavandi, Saeid, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, vol. 76, pp. 243–297, 2021, ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2021.05.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253521001081.

[190] Barraquand, Jerome and Latombe, Jean-Claude, "Robot motion planning: A distributed representation approach," *The International Journal of Robotics Research*, vol. 10, no. 6, pp. 628–649, 1991.

[191] Fridovich-Keil, David, Bajcsy, Andrea, Fisac, Jaime F., Herbert, Sylvia L., Wang, Steven, Dragan, Anca D., and Tomlin, Claire J., "Confidence-aware motion prediction for real-time collision avoidance1," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 250–265, 2020. DOI: 10.1177/0278364919859436. [Online]. Available: https://doi.org/10.1177/0278364919859436.

[192] Malone, Nick, Chiang, Hao-Tien, Lesser, Kendra, Oishi, Meeko, and Tapia, Lydia, "Hybrid dynamic moving obstacle avoidance using a stochastic reachable set-based potential field," *IEEE Tran. Robotics*, vol. PP, pp. 1–15, Jun. 2017. DOI: 10.1109/TRO.2017.2705034.

[193] Nam, Yun Seok, Lee, Bum Hee, and Kim, Moon Sang, "View-time based moving obstacle avoidance using stochastic prediction of obstacle motion," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, 1996, 1081–1086 vol.2. DOI: 10.1109/ROBOT.1996.506852.

[194] Brunton, Steven L., Brunton, Bingni W., Proctor, Joshua L., Kaiser, Eurika, and Kutz, Nathan J., "Chaos as an intermittently forced linear system," *Nature Communications*, vol. 8, no. 1, 2017. DOI: 10.1038/s41467-017-00030-8.

[195] Champion, Kathleen, Lusch, Bethany, Kutz, J. Nathan, and Brunton, Steven L., "Data-driven discovery of coordinates and governing equations," *Proceedings of the National Academy of Sciences*, vol. 116, no. 45, pp. 22 445–22 451, 2019. DOI: 10.1073/pnas.1906995116.

[196] Kamb, Mason, Kaiser, Eurika, Brunton, Steven L., and Kutz, J. Nathan, "Time-delay observables for koopman: Theory and applications," *SIAM Journal on Applied Dynamical Systems*, vol. 19, no. 2, pp. 886–917, 2020.

[197] Juang, Jer-Nan and Pappa, Richard S., "An eigensystem realization algorithm for modal parameter identification and model reduction," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 5, pp. 620–627, 1985. doi: 10.2514/3.20031.

[198] Hassani, Hossein and Mahmoudvand, Rahim, "Multivariate singular spectrum analysis: A general view and new vector forecasting approach," *Int. J. Energy & Statistics*, vol. 01, Apr. 2013. doi: 10.1142/S2335680413500051.

[199] Damen, A. A. H., Hof, Van den, and Hajdasiński, A. K., "The page matrix : An excellent tool for noise filtering of markov parameters, order testing and realization," 1982. [Online]. Available: https://api.semanticscholar.org/CorpusID:122758913.

[200] Gavish, Matan and Donoho, David L., "The optimal hard threshold for singular values is $4/\sqrt{3}$," *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 5040–5053, 2014. doi: 10.1109/TIT.2014.2323359.

[201] Hirsh, Seth M., Ichinaga, Sara M., Brunton, Steven L., Nathan Kutz, J., and Brunton, Bingni W., "Structured time-delay models for dynamical systems with connections to frenet–serret frame," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 477, no. 2254, p. 20 210 097, 2021. doi: 10.1098/rspa.2021.0097.

[202] Tordesillas, Jesus and How, Jonathan P., "Mader: Trajectory planner in multiagent and dynamic environments," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 463–476, 2022. doi: 10.1109/TRO.2021.3080235.

[203] Fawcett, Randall T., Afsari, Kereshmeh, Ames, Aaron D., and Hamed, Kaveh Akbari, "Toward a data-driven template model for quadrupedal locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7636–7643, 2022. doi: 10.1109/LRA.2022.3184007.

[204] Stellato, B., Banjac, G., Goulart, P., Bemporad, A., and Boyd, S., "OSQP: An operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020. doi: 10.1007/s12532-020-00179-2. [Online]. Available: https://doi.org/10.1007/s12532-020-00179-2.