STOCHASTIC CONTROL VIA CHANCE CONSTRAINED OPTIMIZATION AND ITS APPLICATION TO UNMANNED AERIAL VEHICLES

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Michael P. Vitus

August 2012

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Claire J. Tomlin) Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Stephen M. Rock)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Stephen P. Boyd)

Approved for the University Committee on Graduate Studies

Abstract

Automation has had a strong presence in manufacturing for centuries, however its use in society has been very limited. The most ubiquitous robot in everyday life is the Roomba which has sold over 6 million, but the vacuuming task of this robot is simple in comparison with the difficult tasks performed in the manufacturing sector. Automation is successful in manufacturing because the task and environment can be well-defined to simplify the problem, whereas for automation to penetrate into society, it needs to safely operate in the presence of significantly greater uncertainty.

Applications that could benefit from increased autonomy include: robot-assisted surgery, energy efficient control of buildings, autonomous control of vehicles, robotic assistance for elderly and disabled people, routing aircraft around weather, and home automation for tasks such as folding laundry, loading and emptying a dish washer, or tidying up a room.

All of the above applications are examples of stochastic systems which contain three sources of uncertainty: process uncertainty, sensing uncertainty, and environment uncertainty. In order to safely operate in the intended domains, the system must account for all three types of uncertainty in generating a safe control strategy. This problem of generating control inputs for systems under uncertainty is commonly referred to as the stochastic control problem. One way of formulating this problem is as a chance constrained optimization problem that restricts the risk of violating the system's constraints to be below a user supplied threshold.

This thesis develops several extensions over existing chance constrained programming solutions. The feedback controller is used to shape the uncertainty of the system to facilitate the satisfaction of the stochastic constraints, enabling previously infeasible solutions. Systems with component failures are also studied, and the computational complexity is drastically reduced over previous solution methods. The chance constrained framework is also extended to handle systems operating in uncertain environments. A novel hybrid approach is developed that uses a combination of sampling and analytic functions to represent the uncertainty. This approach results in a convex optimization program, guaranteeing the optimal solution and reducing the complexity over other methods. The formulation is further extended to incorporate future measurements of the uncertain environment to increase the performance of the system.

The proposed stochastic control methods are solved in real-time to plan trajectories for a quadrotor unmanned aerial vehicle navigating in a three-dimensional cluttered, uncertain environment. The solution method enables the quadrotor to explore the environment to gather more information, allowing it to successfully complete its objective.

Acknowledgement

If I have seen further it is by standing on the shoulders of giants.
Isaac Newton

During my PhD I have been impacted by many people and I am indebted to them for their wonderful support. First and foremost, I want to express sincere gratitude to my advisor, Professor Claire Tomlin. The combination of her support and enthusiasm has created an exceptional lab environment that allowed me freedom to explore my interests. Her guidance throughout the years has been invaluable in helping me grow not only professionally but personally as well. It never ceases to amaze me all the little things she did to ensure my graduate life was fulfilling and enjoyable.

I would also like to thank all of my committee members: Professors Stephen Boyd, Chris Gerdes, Jean-Claude Latombe, and Steve Rock for providing me with valuable feedback that has impacted every aspect of this thesis. In addition, they are fantastic teachers who have inspired me to never stop learning. I especially want to thank Professors Steve Rock, Stephen Boyd and Jean-Claude Latombe. Steve Rock has always been a significant source of motivation and I have enjoyed our wonderful talks. I am very grateful for how he made me feel like an honarary member of the Aerospace Robotics Lab. Stephen Boyd is a constant source of information and I thank him for many stimulating and enriching discussions. I am amazed at the wealth of information that he can provide on any topic. In addition to being a wonderful professor, Jean-Claude Latombe has been one of the most courteous and considerate people I know. He has constantly provided me with invaluable feedback, and I have always been impressed with his ability to instantly grasp a new problem and offer significant insight into potential solutions.

I am extremely grateful for having overlapped in the lab with Gabe Hoffmann, Vijay Pradeep, Kaushik Roy, and Steve Waslander. They not only helped create an enjoyable working experience, but they were instrumental in shaping me as a researcher. I owe a special thank you to Haomiao Huang who I have had the distinct pleasure of undertaking this journey with from start to finish. I thank him for being a great colleague and a good friend. I was also extremely fortunate to have the opportunity to collaborate and become friends with Wei Zhang. He is a constant source of happiness and I am thankful for his mentoring during the end of my PhD. Finally, I wish to thank everyone in the Aerospace Robotics Lab for frequent discussions and for creating a fun and enjoyable working environment.

In my wildest dreams, I never imagined that I would find a second family while in graduate school. I owe enormous thanks to the amazing group of friends I have found here that provided a constant source of happiness for me: Sean Augenstein, Steve Pifko, Lukas and Vanessa Kuhn, Matt Simms, Sarah Garrett, Ronan Flanagan, Monika Jedrzejowska, San Gunawardana, Alan and Molly Zorn, Kelley and Josh Alwood, Dan and Maureen Berkenstock, Mike Trela, Jeremy and Kara Harris, Andrew Smith, Joelle Faulkner, Eric Doran, Gene Lee, and Gwen Yoshinaga.

My family has provided unwavering support during this experience and it is impossible to convey how much they mean to me. I wish to express enormous gratitude to my parents Joseph and Janet, my siblings Kim, Bobby, and Joey, and my nieces and nephew Melanie, Bobby, Katy, Calynn, and Bryana.

Finally, I would like to offer the greatest thanks to my girlfriend and best friend Debbie Meduna. She never ceases to amaze me with her ability to make me smile and laugh which has carried me through this process. She has been my source of motivation during my PhD and I am glad I was able to make this journey with her. I look forward to our many new exciting adventures together!

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis presents stochastic control methods to enable robust operation of autonomous systems under uncertainty. The primary goal of this thesis is to increase the applicability of current stochastic control methods to include optimization over the feedback controller, predictive control of systems with component failures, treatment of environmental uncertainty, and incorporation of future environmental sensing. In addition, a thorough comparison of different stochastic constraint handling methods is presented, providing the criteria to make an informed decision of the methodology to use for a given application.

1.1 Overview

The primary motivating application for this thesis is stochastic motion planning for robotic systems. Robotic and autonomous systems are becoming increasingly prevalent in everyday life; there are robots which clean floors, teleprescence robots that can buy and deliver breakfast, personal robots for cleaning a room or emptying a dishwasher, robotic surgeons, driver assistance systems for pedestrian detection and lane changing, and even fully autonomous cars. A critical challenge for control and planning in these systems is the presence of uncertainty. Other planning applications that share this challenge include air traffic control around weather, chemical process control, energy efficient control of buildings and electric cars, and financial engineering. For all of these examples, there is a pressing need to develop robust and safe algorithms in the presence of uncertainty.

All of the above applications are examples of stochastic systems, where for the purposes of this thesis, a stochastic system is defined as a process containing the following sources of uncertainty:

- process uncertainty
- sensing uncertainty
- environment uncertainty.

Process uncertainty refers to disturbances that cause the system to deviate from the expected trajectory. Examples of disturbances accounted for by process uncertainty include wind, varying surface properties over which a vehicle might be traveling, tissue deformation for needle insertion, discrepancies in building occupancy, or unmodeled dynamics. Sensing uncertainty refers to the inability to sense the system state exactly, due to noise in the sensors or the partial observability of the system. This will act as a disturbance when trying to control the system. Environmental uncertainty refers to uncertain events in the surrounding environment, outside of the system of interest. Examples of environmental uncertainty are: natural disasters that cause drastic changes to the structure of the environment, uncertainty in the behavior of humans or other autonomous agents, or weather forecast uncertainty for building control.

The presence of these uncertainties means that the exact system state is never truly known. There are many different techniques to address this problem. The simplest approach employs deterministic methods and accounts for the system disturbances by adding buffers to the constraints. However, this approach typically results in a high risk of failure because the buffer sizes are not determined using the distribution of the uncertainties. Another approach is to use robust control methods which account for system uncertainties through worst case bounds on the system state, formulated based upon bounds on the disturbances. For safety critical systems these methods are valuable, but in applications that can tolerate failures the robust control solutions tend to be conservative. For the applications of concern in this thesis, a detailed model of the system's dynamics is available, allowing the distribution of the system's uncertainty to be exploited in the control solution, generally leading to increased system performance over these other approaches.

For these stochastic control methods, the problem is solved in the space of probability distributions of the system state, defined as the belief space. However, control and planning in the belief space does not guarantee success because there might be a small probability that a large disturbance would occur causing the system to violate its constraints. Therefore, any stochastic control solution must trade off between the conservativeness of the plan and the performance of the system.

To use the stochastic control methods developed in this thesis, a description of the uncertainty of the system is required. Fortunately, for the motivating applications, the necessary modeling has often already been performed, and in many of these cases the systems are described well by a linear Gaussian process (one such system is used in the experimental demonstration in Chapter 9). In cases where linear Gaussian models are inaccurate, the distribution of the system can be represented using sampling methods as used in Chapter 6. If only an approximate description of the uncertainty can be obtained, the methods in this thesis are still applicable and will provide a valuable heuristic method but there is no formal guarantee that the constraints will be satisfied.

1.2 Motivating Applications

The extensions of the stochastic control framework developed in this thesis are primarily motivated by the growing need for robust operation of autonomous systems under uncertainty. This section describes several specific examples of autonomous systems that would benefit from using this stochastic control framework. These example applications include robot-assisted surgery, energy efficient control of buildings, vehicle control, and robotics.

1.2.1 Robot-Assisted Surgery

Robot-assisted surgery involves using robotic manipulators during surgery. This technology has the potential to increase the effectiveness of surgical procedures. Specifically, robot-assisted surgery can improve minimally invasive surgery, decrease incision sizes which reduces patient recovery times, and improve the surgeon's control of surgical instruments [1].

One task that is well suited for robotic surgery is needle insertion through soft tissue. Needle insertion is used in numerous medical procedures including biopsy to obtain a specific tissue sample for testing, drug injections for anesthesia, and radioactive seed implantation for brachytherapy cancer treatment [2]. Recent results suggest that steerable needles will enhance targeting accuracy while avoiding critical areas in the human body [3].

A steerable needle has a bevel tip resulting in asymmetric forces when traveling through the tissue. These asymmetric forces can be leveraged to help guide the needle to the desired location. The steerable needle can be controlled by applying a constant rotation or a constant linear velocity. However, the motion of the needle through the tissue causes deformations resulting in large disturbances to the motion of the needle. These disturbances can cause the needle to enter into critical regions and cause damage to organs.

To increase the performance of the steerable needles, it has been proposed to automate the process [4], and this automation fits naturally in the proposed stochastic control framework. By employing this framework, the disturbances on the motion of the needle can be directly accounted for. This reduces the risk of damaging organs or bones and increases the accuracy of the procedure.

1.2.2 Energy Efficient Control of Buildings

Consumption of energy and other natural resources is increasing at an alarming rate which must be reduced to create a sustainable future [5]. One sector where a large savings of energy can be realized is the buildings sector, accounting for 40% of the energy consumption and 40% of the greenhouse gas emissions in the United States [6].

Heating, ventilation, and air conditioning (HVAC) systems account for 33% of the buildings' energy usage [7], and typically employ simple on-off controllers. Consequently, the simplest way to reduce the energy consumption of buildings, without having to modify the existing infrastructure, is to alter this existing control strategy.

The control of a building's HVAC system presents a significant challenge due to the uncertainty in the room occupation, process dynamics, and weather (temperature and solar radiation) which can create large disturbances. The control system is tasked with keeping the temperature within a specified comfort zone, regardless of these disturbances. However, the building standards do not require that these temperature constraints always be satisfied [8].

These soft temperature constraints as well as the types of uncertainties present can be directly accounted for in the proposed stochastic control framework. By exploiting a small probability of temperature constraint violation, the energy consumption can be significantly reduced. Such control methods have been shown by Oldewurtel et al. [8] to have the potential to drastically reduce the energy consumption in buildings.

1.2.3 Vehicle Control

In the near future, autonomous vehicles will be commonly used in our everyday lives. Before fully autonomous vehicles are developed, assistance systems used to aid the driver will grow in complexity. These systems have the potential to increase the safety of the car system as well as increase the capacity of the roadways while simultaneously decreasing the environmental impact. However, in order to safely integrate these systems into mainstream driving, they must account for the underlying uncertainty in the driving process. This uncertainty stems from sensing errors, motion noise due to varying road properties, component failures, and the unknown intent of the other drivers (or autonomous systems). The presence of these uncertainties introduces some level of risk for drivers.

One example that illustrates the risk that drivers assume is when passing a tractor trailer. On all sides of the tractor trailer there are large blind spots that prevent the driver from seeing other vehicles. According to the California driver's handbook, the driver should not remain along side of it and should instead pass it as quickly as possible. This policy minimizes the risk of the tractor trailer colliding with the driver if the car is in the blindspot of the tractor trailer.

Fully autonomous cars must also assume some risk and generate plans taking into account the uncertainty in the behavior of other drivers or autonomous vehicles. This type of planning problem is easily formulated by the proposed stochastic control framework. Using the probability distributions of the uncertainties in the system, the framework can safely enable various planning tasks such as passing other vehicles, changing lanes, and entering intersections.

1.2.4 Robotics

There are many applications for the proposed stochastic control and planning framework in the field of robotics. One potential application is simultaneous localization and mapping (SLAM) which is a fundamental task in robotics [9]. SLAM is concerned with constructing a globally consistent model of the environment and consists of a vehicle navigating through and sensing an unknown environment. Typical approaches are only concerned with the estimation process and the robot is manually guided through the unknown environment. These approaches, however, neglect how the control inputs and/or path for the robot affects the quality of the map, even though it can have a dramatic effect.

Another application that has garnered attention from the robotics community is in home automation for tasks such as: folding laundry, loading and emptying a dishwasher, or tidying up a room. In all of these examples, robot manipulation plays a key role in the successful completion of the tasks. However, there can be a large amount of uncertainty in how the manipulator interacts with objects in the environment. For instance, in trying to pick up an object, the friction properties, the rigidity of the object, the weight distribution, and the estimate of the location will all affect the outcome.

The other application is in robot assisted herding [10], which is concerned with corralling a group of animals. The difficulty in this scenario is incorporating the stochastic nature of the animals as well as with the human/computer coordination.

For all of the above example applications, the proposed framework will allow the system to safely operate in the presence of uncertainty and to probe the surroundings to gather information to help successfully complete the task.

1.3 Stochastic Optimization

This thesis considers solutions to the stochastic optimization program presented in this section. First, consider the following standard optimization program

$$\begin{array}{ll} \text{minimize} & \phi(x,\xi) \\ \text{subject to} & \\ & \psi\left(x,\xi\right) \leq 0 \end{array} \tag{1.1}$$

where $\xi \in \mathbb{R}^k$ is a parameter vector and $x \in \mathbb{R}^n$ is a vector of variables. If ϕ and ψ are convex functions, then the optimization program (1.1) is convex and can be solved efficiently [11]. In many practical applications, however, the parameter vector ξ may be uncertain causing the previous optimization program to be ill-defined. One way of handling the uncertainty is to explicitly treat the constraints probabilistically, leading to the following stochastic optimization program.

$$\begin{array}{ll} \text{minimize} & \mathbf{E}\left[\phi(x,\xi)\right] \\ \text{subject to} & \\ & \mathrm{P}\left(\psi\left(x,\xi\right) \leq 0\right) \geq 1-\delta \end{array} \tag{1.2}$$

In this formulation, the probability of constraint satisfaction is restricted to be larger than $1-\delta$ (where $0 \geq \delta < 1$). This probabilistic constraint is referred to as a chance constraint. Depending on whether ψ is scalar-valued or vector-valued, the constraint is referred to as an individual or joint chance constraint respectively.

The choice of δ for the chance constraint is application specific, and needs to be determined by the user from the problem specifications. Typically, the applications considered in this thesis are concerned with values of δ of 0.01, 0.05, or 0.1. If

δ is smaller, then a very accurate distribution of the tails is required, otherwise the constraints will not hold during execution. For the simulation examples in this thesis, a wide range of δ values is used in order to highlight differences between the various presented methods.

Chance constrained optimization was first introduced by Charnes, Cooper and Symonds in 1958 [12], Miller and Wagner in 1965 [13], and Prékopa in 1970 [14]. There are two fundamental difficulties in solving the chance constrained optimization program. Even if the original program (1.1) were convex, there is no guarantee that the program (1.2) is convex, which could make the optimization difficult. The second complication is in evaluating the constraint violation probability. Typically, this requires the calculation of a multivariate integral which generally does not have a closed form, analytical solution.

Convexity of the Chance Constrained Optimization Program

The optimization program (1.2) was shown to be convex in [15, 16] for an individual, affine constraint of the form $\psi(x, \xi) = \xi^{\mathrm{T}} \begin{bmatrix} x \\ 1 \end{bmatrix}$ if ξ is Gaussian. An extension of this result was given by Lagoa et al. [17] for the case in which ξ has a log-concave and symmetric density. Prékopa [18] showed that if $\psi(x, \xi)$ is quasiconvex in (x, ξ) and if ξ has a log-concave probability distribution, then the resulting program is convex. Two examples of log-concave probability distributions are the uniform and multivariate Gaussian distributions. For the case of individual chance constraints of the form $\mathrm{P}(\xi^{\mathrm{T}} q(x) \leq b) \geq 1 - \delta$, Henrion [19] showed that the constraints form a convex set if $q(x)$ is nonnegative and convex, ξ has an elliptically symmetric distribution, and $\delta < 0.5$.

Approximation/Bounding Approaches

In the case that the convexity of the optimization program (1.2) cannot be guaranteed, approximation methods can be employed which bound the probability distribution, leading to feasible and conservative solutions to the original problem. Ben-Tal and Nemirovski [20] proposed a quadratic approximation, Rockafellar and Uryasev [21] proposed the conditional value-at-risk (CVaR) approximation, and Nemirovski and Shapiro [22] proposed the Bernstein approximation. The latter two methods replace the indicator function in the integral for calculating the probability by a convex bounding function. Hong et al. [23] extended the CVaR approach by solving a series of convex approximations that iteratively tightens the convex bounding function. This method was shown to converge to a Karush-Kuhn-Tucker point of the original chance constrained problem.

The work by van Hessem et al. [24, 25, 26, 27, 28] studied the chance constrained closed-loop model predictive control problem with Gaussian noise disturbances. They optimized over the feedback control laws and open-loop inputs while ensuring that the chance constraints on the overall system were satisfied. To convert the stochastic problem into a deterministic one, they developed an ellipsoidal set bounding approach that results in a set of second order cone constraints, however this led to a conservative solution. This work was extended by Blackmore [29] to handle nonconvex feasible regions. Shin and Primbs [30] also used the ellipsoidal set bounding approach and proposed an efficient interior point method that exploited the Riccati structure of the problem to decrease the computational complexity.

The work by Blackmore et al. [31] uses the work presented in [32] to bound the joint chance constraints with Gaussian noise using Boole's inequality, which typically leads to a very small amount of over-conservativeness. They also used the idea of risk allocation introduced by [32] to distribute the risk of violating each chance constraint while still guaranteeing the specified level of safety. By using the risk allocation technique instead of assuming a constant amount of risk for each constraint, the performance of the overall system can be significantly increased. Ono and Williams [33] extended this method to multi-agent system, proposing a decentralized algorithm for solving the chance constrained optimal control problem.

Another approximation used in solving the optimization program (1.2) is the scenario approach which draws samples for the uncertain parameters and requires the constraints to be satisfied for each sample. Given the convexity of the original optimization program (1.1), this method also results in a convex program. In this approach, the number of samples required needs to be determined in order for the original chance constraints to hold with a large probability. This problem was studied by Calafiore and Campi [34] as well as De Fairas and Van Roy [35]. This method can result in very conservative solutions or infeasible programs, limiting its applicability. Luedtke and Ahmed [36] proposed the sampled approximation technique, an extension to the scenario approach, that allows the choice of which sampled constraints to satisfy.

Ruszczyński [37] and Blackmore [38] proposed another sampling approach that draws samples from the uncertainty distributions and uses binary variables to count the number of constraint violations. This transforms the original stochastic control problem into a deterministic mixed integer program. This sampling approach, however, becomes intractable as the number of samples needed to accurately represent the true belief state increases. Platt and Tedrake [39] extend the sampling and mixed-integer programming formulation to solve a class of partially observable problems with state dependent noise.

The optimization problem with uncertain parameters can alternatively be handled by robust techniques [40], [41], [42], [43], [44]. Rather than using chance constraints, these methods assume bounds on the unknown parameters which can be used to formulate worst case bounds on the system state. However, such an approach is conservative because it disregards the information that is often available about the distribution of the system's uncertainty.

Researchers [45], [46], [47] have also developed lower bounds for the stochastic control problem, which are useful for comparing the performance of the approximation methods.

Stochastic Motion Planning in Robotics

The application of the stochastic optimization problem in Equation (1.2) has many motivating applications in robotics, primarily in stochastic motion planning. For deterministic motion planning, incremental sampling-based algorithms, such as Probabilistic Roadmaps (PRM) [48] or Rapidly-Exploring Randomized Trees (RRT) [49], are the most popular algorithms due to their ability to handle large dimensional systems and low computational complexity. These methods construct graphs of feasible paths or trajectories through the environment and search the graph for a solution to the goal. Recently, Karaman and Frazzoli [50] have proposed RRT* which converges to the optimal solution almost surely. Given the effectiveness of the incremental sampling-based algorithms for deterministic motion planning, several researchers have extended them to solve the stochastic motion planning problem.

Prentice and Roy [51] proposed an extension called the Belief Roadmap (BRM) to efficiently plan in the belief space. For a linear Gaussian system, they devised a factorization of the covariance matrix from a Kalman filter estimator which allows several prediction and measurement steps to be combined into a single linear transfer function. Using this transfer function, they can efficiently search through a graph of the environment for the minimum covariance path to reach the goal. However, they do not solve for the control inputs for the system to follow the path. A Markov Decision Process (MDP) based algorithm was proposed by Alterovitz et al. [52] that accounts for the motion uncertainty of the system but does not account for the partial observability of the system state or the sensing uncertainty.

Others [53], [54], [55] [56] have included both the motion and sensing uncertainty when planning trajectories through the environment, but they simplify the problem by assuming the maximum likelihood observation is received for all future time-steps. This approximation results in an inaccurate representation of the probability distribution of the state which can lead to a violation of the constraints on the system. To better represent the system state's distribution, Bry and Roy [57] calculated the distribution of the closed-loop state based upon the distribution of the observations. They used this distribution along with an extension of the RRT* algorithm to check for constraint violations.

Van den Berg et al. [58] characterized the probability distribution of a system based upon a linear quadratic regulator controller with Gaussian models of the process and measurement noises. They proposed a two step planning process named LQG-MP: (1) a set of candidate paths were generated without accounting for the system uncertainty, (2) the best path was selected based upon a criterion which incorporated the uncertainty of the state. Since the uncertainty wasn't taken into account in generating the paths, the paths generated were often suboptimal solutions.

Patil et al. [59] provided an extension of the LQG-MP for motion planning in highly deformable environments with application to computer-assisted surgery. To improve the estimated probability distributions, they used a simulator and optimal control to numerically estimate the time-dependent linear dynamics for the uncertain motion through the deformable materials.

Incorporating environment uncertainty into the motion planning problem has also received some attention. Missiuro and Roy [60] handled uncertain environments by modifying the sampler used in a probabilistic roadmap. However, while the algorithm accounts for the environment uncertainty, motion noise or sensing noise is not accounted for. Burns and Brock [61] also proposed a sampling-based planner to incorporate environment sensing uncertainty into the planning process but do not directly handle motion noise. Stachniss et al. [62] employed a decision-theoretic framework to choose between macro actions to reduce the uncertainty of the environment during exploration.

A general principled way of modeling the problem of stochastic motion planning through uncertain environments is by using Partially Observable Markov Decision Processes (POMDPs) [63]. A POMDP models a system's decision process in which the system's dynamics are not necessarily deterministic, but rather the outcome of its actions could be stochastic, and the system does not know the true underlying state. Instead, the system must take measurements of the underlying state and maintain a probability distribution over all possible states. In order to solve a POMDP, a policy must be calculated for the entire belief space that determines the best action for every possible future belief state that it may encounter. The main difficulty in solving a POMDP is its large computational complexity, and in general solving a POMDP exactly is intractable. The two main causes of the large computational complexity is the curse of dimensionality and the long planning horizons required to generate useful plans.

Several researchers have proposed efficient solution techniques for solving POMDPs. The work of Hsu et al. [64, 65] proposed two ideas to decrease the computational complexity and exploit the efficiency of point-based POMDP solvers. They utilized the mixed observability of the problem by factoring the state into two subsets of fully and partially observable states. For this representation, they only needed to maintain a belief space over the partially observable subset. They also proposed a milestone guided sampling algorithm to guide the sampling of the point-based POMDP solvers. The use of macro-actions has also been used to decrease the computational complexity for planning and learning algorithms [66, 67, 68, 69]. Instead of using the full action space, these methods use a sequence of primitive actions called macro-actions; this restricts the policy space and reduces the action branching factor leading to longer planning horizons.

1.4 Limitations of Existing Methods

Despite the progress in the field of stochastic control, there are several limitations which prevent direct application of these methods for the motivating applications of this thesis. Three of these limitations that are addressed in this thesis are presented below.

1. *Computational efficiency restricts real-time applicability.*

   Many of the proposed solution methods for the stochastic control problem are not concerned with computational efficiency. However, this restricts their applicability for online control, which is required for many of the motivating applications.

2. *Conservative solution for feedback design.*

   The previously proposed methodology for optimizing over the feedback controller uses a conservative bounding approach and assumes a fixed risk for the constraints. This causes the method to be overly conservative, resulting in the elimination of feasible solutions.

3. *Current methods do not consider both uncertain constraint parameters and variables.*

   Previous methods have not considered the case in which both the constraint parameters and variables are uncertain. This typically occurs in many of the motivating applications for the proposed stochastic control framework. Furthermore, the previous methods cannot incorporate future constraint parameter sensing into the formulation, resulting in inefficient solutions or infeasible problems.

1.5 Contributions

This research addresses the limitations highlighted in the previous section to enable the application of stochastic control for the motivating problems. This resulted in several contributions to the field of stochastic control:

1. *A comparison of different methods to handle the chance constraints.*
   A rigorous evaluation of stochastic constraint handling methods is conducted, illustrating the trade-offs of each technique and providing clear criteria for making an informed decision for a given application.

2. *Reduced conservativeness for feedback design.*
   In order to reduce the conservativeness of feedback design in the stochastic control problem, a novel method is developed to optimize over both the risk allocation and the feedback controller. The method is shown to enable previous infeasible solutions.

3. *Improved computational efficiency of stochastic control for systems with component failures.*
   By applying the methods presented in this thesis, the computational efficiency for this problem is drastically reduced as compared with previous solution methodologies.

4. *Extended the chance constrained framework to handle problems with both uncertain constraint parameters and variables.*
   The stochastic control framework is extended to handle uncertainty in both constraint parameters and variables. A novel hybrid approach is developed to

solve this extended problem, which uses a combination of sampling and analytic functions to represent the probability distributions of the system and constraint parameters. This approach results in a convex optimization program under certain conditions, guaranteeing the optimal solution and reducing the computational complexity.

5. *Extended the chance constrained problem formulation to incorporate new measurements of the uncertain constraint parameters.*
   A novel hierarchical method is developed to incorporate new measurements of the constraint parameters. This method splits the stochastic control problem into two stages: (i) an exploration stage that calculates a high level policy for incorporating information from potential measurements, (ii) an execution stage that solves a stochastic control problem to execute the high level policy.

6. *First successful, real-time experimental demonstration of chance constrained control with uncertain constraint parameters and variables.*
   The proposed stochastic control methods are solved in real-time to plan trajectories for a quadrotor unmanned aerial vehicle navigating in a three-dimensional cluttered, uncertain environment. The solution method enables the quadrotor to explore the environment to gather more information, allowing it to successfully complete its objective.

1.6 Outline

This thesis proceeds in Chapter 2 with the formal definition of the stochastic control problem and is formulated as a chance constrained optimization program. Chapter 3 compares several different methods of handling the chance constraints. The remaining chapters investigate the solution of the chance constrained optimization program for increasingly difficult scenarios.

Chapter 4 considers the simplest case with deterministic constraint parameters and a system that has a fixed controller and estimator, and is based upon the publication [70]. This problem is further extended in Chapter 5 by optimizing over the feedback controller in order to shape the system uncertainty to enable previously infeasible solutions; this chapter is based upon [71]. In Chapter 6, the stochastic control problem for jump Markov linear systems is investigated. This presents several difficulties because the uncertainty in the system is non-Gaussian and thus is difficult to represent analytically. The next two chapters consider the solution of the chance constrained optimization program with uncertain constraint parameters.

Chapter 7 presents a novel hybrid algorithm to solve the chance constrained optimization program with uncertain constraint parameters, which is an extension of previous work [72]. To incorporate sensing of the uncertain constraint parameters into the problem solution, Chapter 8 presents a hierarchical algorithm for solving the dual control problem. Chapter 9 provides an experimental demonstration of the hierarchical algorithm for planning trajectories of a quadrotor unmanned aerial vehicle navigating in a cluttered three-dimensional environment. Chapters 8 and 9 are based upon a forthcoming publication [73].

The thesis concludes in Chapter 10 with a discussion of the contributions of this thesis and directions of future work.

Chapter 2

Stochastic Control Problem

This chapter formulates the stochastic control problem addressed in this thesis. Consider the following discrete-time, finite-time horizon, stochastic system defined by

$$x_{k+1} = f(x_k, u_k, w_k), \quad k = 0, \ldots, N-1, \tag{2.1}$$

where $x_k \in \mathbb{R}^n$ is the system state, $u_k \in \mathbb{R}^m$ is the control input, $w_k \in \mathbb{R}^r$ is the process noise and N is the time horizon. The initial state, x_0, is assumed to be uncertain with a known probability distribution $p(x_0)$. At each time step, a noisy measurement of the state is taken, defined by

$$y_k = h(x_k, v_k), \quad k = 0, \ldots, N-1 \tag{2.2}$$

where $y_k \in \mathbb{R}^p$ and $v_k \in \mathbb{R}^s$ are the measurement output and noise of the sensor at time k, respectively. The process and measurement noise are assumed to have known probability distributions $p(w_k)$ and $p(v_k)$, respectively. These distributions are assumed to be mutually independent at different time-steps. In addition, the process noise, measurement noise and initial state are assumed to be mutually independent.

In the following chapters, it is assumed that a linear description of the system can be obtained. For a nonlinear system, this is achieved by linearizing the system around a nominal trajectory. For this choice, Eqns. (2.1), (2.2) simplify to the following linear system

$$x_{k+1} = A_k x_k + B_k u_k + W_k w_k, \quad k = 0, \ldots, N-1, \tag{2.3}$$

$$y_k = C_k x_k + V_k v_k, \quad k = 0, \ldots, N-1. \tag{2.4}$$

For notational convenience, the system state, measurements, control inputs, process noise and measurement noise for all time-steps are concatenated to form:

$$\mathbb{X} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_N \end{bmatrix} \quad \mathbb{Y} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{bmatrix} \quad \mathbb{U} = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} \quad \mathbb{W} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{N-1} \end{bmatrix} \quad \mathbb{V} = \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{N-1} \end{bmatrix}. \tag{2.5}$$

In an abuse of notation, $\mathbb{X} = f(x_0, \mathbb{U}, \mathbb{W})$ and $\mathbb{Y} = h(\mathbb{X}, \mathbb{V})$ will be used as a compact representation for the calculation of all states and measurements.

Since the process and measurement noise are random variables, the state trajectory and control inputs are also random variables. Consequently the performance of the system must be judged based upon the expected value of an objective function $\phi : \mathbb{R}^{nN} \times \mathbb{R}^{mN} \to \mathbb{R}$, over $\mathbb{X}$ and $\mathbb{U}$:

$$\mathbf{E}\left[\phi(\mathbb{X}, \mathbb{U})\right], \tag{2.6}$$

where the expectation is over $\mathbb{W}$ and $\mathbb{V}$. The objective function is assumed to be a convex function of $\mathbb{X}$ and $\mathbb{U}$.

There is also a set of constraint functions on the state trajectory and control inputs, $\psi_i : \mathbb{R}^{nN} \times \mathbb{R}^{mN} \to \mathbb{R}^{q_i}$, $i = 1, \ldots, N_c$. The constraints functions are assumed to be convex functions. Unfortunately, due to the stochasticity of the problem, the system constraints cannot be considered deterministically; the stochasticity may result in a non-zero chance that the constraints will be violated. Consequently, the constraints must be considered probabilistically, leading to a notion of risk.

Risk is associated with having to make a decision with only partial information due to the future uncertainty in the system, but with the knowledge that this might lead to the violation of some of the constraints in the system. A measure of the risk thus must be introduced to determine the overall cost of violating the constraints. There are several different measures that can be considered. The expected value of the constraints can be used,

$$\mathbf{E}\left[\psi_i\left(\mathbb{X},\mathbb{U}\right)\right] \leq 0, \quad i = 1,\ldots,N_c, \tag{2.7}$$

the worst case of the constraints could be considered (if the process and measurement noise are bounded)

$$\sup_{\mathbb{W},\mathbb{V}} \psi_i\left(\mathbb{X},\mathbb{U}\right) \leq 0, \quad i = 1,\ldots,N_c, \tag{2.8}$$

or the constraints could be restricted to only hold up to a certain probability limit

$$\mathrm{P}\left(\psi_i\left(\mathbb{X},\mathbb{U}\right) \leq 0\right) \geq 1-\delta_i, \quad i = 1,\ldots,N_c. \tag{2.9}$$

This last measure is typically called a chance constraint.

There is not one “correct” way to handle the constraints for all varying applications, however, this thesis will be concerned with the use of the chance constraints. Note, by choosing the probability limit δ_i carefully, all previous choices can be generated. If $\delta_i = 0.5$ and the constant b_i in $\mathrm{P}\left(\psi_i\left(\mathbb{X},\mathbb{U}\right) \leq b_i\right)$ is chosen appropriately then the expected value of the constraints will hold, or if $\delta_i = 0$ then the constraints will almost surely hold even under the worst case noise (assuming the process and measurements noises are bounded). Also, if $\delta_i = 0.5$ then the median value of the constraint will hold.

Finally, the stochastic control problem can be expressed as

$$\begin{array}{ll} \text{minimize} & \mathbf{E}\left[\phi(\mathbb{X},\mathbb{U})\right] \\ \text{subject to} & \\ & \mathbb{X} = f(x_0,\mathbb{U},\mathbb{W}) \\ & \mathbb{Y} = h(\mathbb{X},\mathbb{V}) \\ & \mathrm{P}\left(\psi_i\left(\mathbb{X},\mathbb{U}\right) \leq 0\right) \geq 1-\delta_i, \quad i = 1,\ldots,N_c \end{array} \tag{2.10}$$

where the expectation is over the noise sources $\mathbb{W}$ and $\mathbb{V}$ and the optimization is over the control policy. In general, this optimization program is nonconvex. In this work, only causal output feedback controllers are considered of the form

$$u_k = \varphi_k \left(y_0, \ldots, y_k\right), \quad k = 0, \ldots, N-1. \tag{2.11}$$

In general, the stochastic control problem in (2.10) cannot be solved for the optimal infinite dimensional control policy, but in the following chapters the control policy will be assumed to be of a given form to obtain a tractable problem. For example, a feedback controller will be assumed to have been designed *a priori* and the desired state trajectory will be the optimization variable. Or the control policy will be assumed to be an affine function of the past measurements where the optimization variables are the feedback gains and the open-loop inputs.

The main challenges to solving the optimization program (2.10) is in characterizing the distribution of the system state and control inputs as well as satisfying the chance constraints on the state and control inputs. The following chapter discusses several approaches of dealing with the chance constraints to form a convex optimization program, and the remaining chapters demonstrate how to formulate solutions to program (2.10) for specific applications.

Chapter 3

Chance Constrained Programming

The most challenging aspect of solving the optimization program (2.10) is in evaluating and satisfying the chance constraints. This chapter discusses two general approaches to this: (i) analytical methods and (ii) sampling methods. The first approach exploits properties of the probability distribution of the system state and control inputs to either convert the chance constraints into a set of deterministic ones or efficiently evaluate the probability of violation. The second approach uses sampling to approximate the probability distribution of the system. There are three different sampling approaches that are commonly employed. The first uses mixed integer programming to approximate the chance constraints by counting the number of constraint violations. The second enforces the constraints for all of the samples and determines the probability that the chance constraints will be satisfied. The third bounds the chance constraints with a convex function and then uses the samples to evaluate the convex bound. In Section 3.3, a comparison of the different approaches is made and the conservatism in each method is illustrated through a simple example. In the final section, a discussion of constraint modeling is presented.

Typically there are two forms of chance constraints:

1. *individual chance constraints* that deal with the satisfaction of each constraint separately, requiring each to be satisfied with probability $1 - \delta_i$

$$\mathrm{P}\left(\psi_i\left(\mathbb{X}, \mathbb{U}\right) \leq 0\right) \geq 1 - \delta_i, \quad i = 1, \ldots, N_c \tag{3.1}$$

where each ψ_i is a scalar function;

2. *joint chance constraints* that deal with the satisfaction of several constraints simultaneously, requiring all to be satisfied with probability $1-\delta_i$

$$\mathrm{P}\left(\psi_{ij}\left(\mathbb{X},\mathbb{U}\right) \leq 0,\ j=1,\ldots,q_i\right) \geq 1-\delta_i, \quad i=1,\ldots,N_c \tag{3.2}$$

where each $\psi_i \in \mathbb{R}^{q_i}$ is a vector valued function. It is assumed that each element in ψ_i has been non-dimensionalized so that the violations are comparable and that the violation of any of the constraints $\psi_{ij}\left(\mathbb{X},\mathbb{U}\right) \leq 0$ is equally undesirable.

Both forms of constraints are important, but typically the joint chance constraints are more applicable to the problems that are considered in the following chapters. The methods presented for joint chance constraints are also applicable to individual chance constraints, therefore, this chapter will discuss the evaluation of the joint chance constraints only.

The following sections assume that the convex constraint functions, ψ_i, are affine functions of the state and control inputs,

$$\psi_i(\mathbb{X},\mathbb{U}) = H_i^{\mathrm{T}}\mathbb{X} + F_i^{\mathrm{T}}\mathbb{U} - b_i, \quad i=1,\ldots,N_c. \tag{3.3}$$

where $H_i \in \mathbb{R}^{nN\times q_i}$, $F_i \in \mathbb{R}^{mN\times q_i}$ and $b_i \in \mathbb{R}^{q_i}$. For Chapters 3-6, the parameters H and b are assumed to be deterministic while in Chapters 7 and 8 they are stochastic. Using the affine constraint functions, the chance constraints are

$$\mathrm{P}\left(H_i^{\mathrm{T}}\mathbb{X} + F_i^{\mathrm{T}}\mathbb{U} - b_i \leq 0\right) \geq 1-\delta_i, \quad i=1,\ldots,N_c. \tag{3.4}$$

This affine assumption is not restrictive for the problems considered because it still allows for rate constraints, control input saturation, model validity ranges, and restrictions on the system from entering unsafe or undesired regions.

The joint chance constraints are difficult to evaluate because they require calculating a multivariate integral which in general can only be performed for low dimensions. Typically, there are two different approaches, analytical and sampling, that are used to efficiently evaluate the chance constraints. Both approaches are discussed in the following sections.

3.1 Analytical Methods for Handling the Chance Constraints

In the following subsections, the methods assume a specific analytical form for the disturbance sources and exploit the properties of the probability distribution function to evaluate the probability of violating the constraints.

3.1.1 Ellipsoidal Relaxation

The method developed in [28, 24, 25, 26, 27] assumes that the first and second moments of the process and measurement noise sources can be estimated numerically from previously collected data. Using the estimated first and second moments, this method approximates the true distribution with a Gaussian distribution. Given this choice, in order to evaluate the chance constraints one needs to integrate a multivariate Gaussian density, which typically does not have an analytic solution. Consequently, an ellipsoidal relaxation method is used to reduce it to a tractable problem. The approach simultaneously optimizes over the shape as well as the center of an ellipsoid to guarantee the satisfaction of the chance constraints. This approach is presented below.

To simplify the following presentation of the ellipsoidal relaxation method, let $z \in \mathbb{R}^{n_z}$ and $z \sim \mathcal{N}(\bar{z}, \Sigma_z)$. The chance constraint $\mathrm{P}\left(h_i^{\mathrm{T}} z \ \leq b_i,\ \forall i\right) \geq 1-\delta$ is equivalent to,

$$\alpha \int_{\mathcal{P}_z} \exp\left(-\frac{1}{2}(z-\bar{z})^{\mathrm{T}}\Sigma_z(z-\bar{z})\right) dz \geq 1-\delta, \tag{3.5}$$

where $\alpha = \dfrac{1}{\sqrt{(2\pi)^n \det(\Sigma_z)}}$ and $\mathcal{P}_z$ is the feasible region for z. There is no straightforward way of handling this integral constraint, therefore the constraint is tightened as follows. If the constraint

$$\bar{z} + \mathcal{E}_r \subset \mathcal{P}_z \tag{3.6}$$

is ensured to be satisfied for the ellipsoid

$$\mathcal{E}_r = \left\{\xi : \xi^{\mathrm{T}}\Sigma_z^{-1}\xi \leq r^2\right\} \tag{3.7}$$

for an appropriately chosen r, then the original constraint $\mathrm{P}\left(h_i^{\mathrm{T}}z \leq b_i,\ \forall i\right) \geq 1-\delta$ is implied by

$$\alpha \int_{\bar{z}+\mathcal{E}_r} \exp\left(-\frac{1}{2}(z-\bar{z})^{\mathrm{T}}\Sigma_z(z-\bar{z})\right) dz \geq 1-\delta. \tag{3.8}$$

After simplifying Eqn. (3.8) to a one-dimensional integral by using standard techniques [74], r is chosen such that

$$\frac{1}{2^{\frac{n_z}{2}}\Gamma\left(\frac{n_z}{2}\right)} \int_0^{r^2} \chi^{n_z/2-1}\exp\left(-\frac{\chi}{2}\right) d\chi = 1-\delta. \tag{3.9}$$

One important property of the solution for r in the previous equation is that for a given allowed probability of failure δ, the necessary confidence radius r increases with the dimension of z. As illustrated below, this will affect the conservatism of the solution for this approach.

Now the original constraint $\mathrm{P}\left(h_i^{\mathrm{T}}z \leq b_i,\ \forall i\right) > 1-\delta$ can be replaced by requiring $\bar{z}+\mathcal{E}_r \subset \mathcal{P}_z$, which for the definition of the ellipsoid $\mathcal{E}_r$ is equivalent to requiring

$$h_i^{\mathrm{T}}\bar{z} + r\sqrt{h_i^{\mathrm{T}}\Sigma_z h_i} \leq b_i, \quad \forall i. \tag{3.10}$$

This simplification guarantees that the joint probability of violating any of the constraints, $h_i^{\mathrm{T}}z \leq b_i$, is less than δ. As the dimension of the state z increases, the back-off parameter r also increases causing the conservatism in the ellipsoidal relaxation method to increase.

By using the ellipsoidal relaxation method, the original chance constraint has been reduced to the second order cone constraint in Eqn. (3.10), simplifying even further to a linear constraint for a fixed Σ_z. This has the potential to reduce the computational complexity of solving the optimization program (2.10). A two dimensional example that illustrates the technique is defined in Example 1 below.

Example 1. *Let $w \in \mathbb{R}^2$ be normally distributed according to $w \sim \mathcal{N}(0, I)$, $x \in \mathbb{R}^2$,*

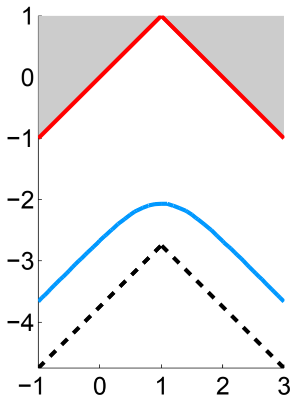


Figure 3.1: An illustration of the ellipsoidal relaxation technique for Example 1. The original constraints are the red solid lines, and the ellipsoidal relaxation technique is shown as the black line. The line that is equal to the desired probability of failure is shown as the blue solid line. The feasible region for the state is below the corresponding lines.

$H = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}$, $b = [0\ 2]^T$, *and finally the joint chance constraint to be satisfied is*

$$\mathrm{P}\left(H^T(x+w) - b \le 0\right) \ge 0.9.$$

The ellipsoidal relaxation method applied to this example is shown in Figure 3.1. The original constraint is the red solid line and the ellipsoidal relaxation method guarantees the satisfaction of the joint chance constraint by restricting the variable's feasible region to be below the black dashed line. The blue solid line is the line that equals the desired probability of failure. As is illustrated, this method tends to be conservative in predicting the feasible region for the system, and the conservativeness increases with the size of the state space due to the resulting increase of the back-off parameter r.

3.1.2 Boole's Inequality

Similar to the ellipsoidal relaxation approach, Boole's method assumes that the process and measurement noise sources are, or can be approximated by, Gaussian random variables. Let $z \in \mathbb{R}^{n_z}$, $z \sim \mathcal{N}\left(\bar{z}, \Sigma_z\right)$, $H \in \mathbb{R}^{n_z \times q}$, and $b \in \mathbb{R}^q$. In general, even for Gaussian distributions, the joint chance constraint $\mathrm{P}\left(H^{\mathrm{T}} z \leq b\right)$ does not have a closed-form analytical expression and requires the evaluation of multivariate integrals.

For Gaussian noise sources, the joint chance constraints can be converted into univariate integrals by using Boole's inequality to conservatively bound the probability of violation. Boole's inequality states that for a countable set of events $E_1, E_2, \ldots$, the probability that at least one event happens is no larger than the sum of the individual probabilities:

$$\mathrm{P}\left(\bigcup_i E_i\right) \leq \sum_i \mathrm{P}\left(E_i\right). \tag{3.11}$$

The chance constraint

$$\mathrm{P}\left(H^{\mathrm{T}} z \leq b\right) \geq 1-\delta \tag{3.12}$$

is equivalent to

$$\mathrm{P}\left(H^{\mathrm{T}} z > b\right) \leq \delta. \tag{3.13}$$

By using Boole's inequality the probability of the constraints being violated is bounded above by

$$\mathrm{P}\left(H^{\mathrm{T}} z > b\right) = \sum_{i=1}^{q} \mathrm{P}(h_i^{\mathrm{T}} \mathbb{X} > b_i). \tag{3.14}$$

Consequently, the multivariate constraints are converted to univariate constraints in Eqn. (3.14), which can be computed through

$$\begin{aligned} \mathrm{P}(h_i^{\mathrm{T}} z > b_i) &= \mathrm{P}(y_i > b_i) \\ &= \frac{1}{\sqrt{2\pi h_i^{\mathrm{T}} \Sigma_z h_i}} \int_{b_i}^{\infty} \exp\left(-\frac{(y_i - h_i^{\mathrm{T}} \bar{z})^2}{2 h_i^{\mathrm{T}} \Sigma_z h_i}\right) dy_i \\ &= \frac{1}{\sqrt{2\pi}} \int_{\frac{b_i - h_i^{\mathrm{T}} \bar{z}}{\sqrt{h_i^{\mathrm{T}} \Sigma_z h_i}}}^{\infty} \exp\left(-\frac{\xi^2}{2}\right) d\xi \\ &= 1 - \Phi\left(\frac{b_i - h_i^{\mathrm{T}} \bar{z}}{\sqrt{h_i^{\mathrm{T}} \Sigma_z h_i}}\right). \end{aligned} \tag{3.15}$$

The function $\Phi(\cdot)$ is the Gaussian cumulative distribution function which does not have an analytic solution, but can be efficiently evaluated using a series approximation or a pre-computed lookup table.

Now that the calculation of a single chance constraint has been developed, the satisfaction of the joint chance constraint, $\mathrm{P}\left(H^{\mathrm{T}} z > b\right) \leq \delta$, needs to be addressed. Previously, two methods have been proposed for this: fixed risk [22] and risk allocation [32, 31].

The fixed risk method assigns a pre-defined allowed probability of violation for each univariate constraint $\mathrm{P}(h_i^{\mathrm{T}} z > b_i)$. This pre-defined probability is chosen to ensure that the total probability of violation is below the threshold δ. Typically a uniform allocation of δ/q is used. Since the allowed risk for each constraint is known beforehand, the chance constraints can be simplified by modifying the system's feasible region at each time-step. For a uniform allocation of risk, this is achieved by requiring the following constraints to hold

$$1 - \Phi\left(\frac{b_i - h_i^{\mathrm{T}} \bar{z}}{\sqrt{h_i^{\mathrm{T}} \Sigma_z h_i}}\right) \leq \frac{\delta}{q}, \quad i = 1, \ldots, q, \tag{3.16}$$

which can be simplified further to

$$h_i^{\mathrm{T}} \bar{z} + \Phi^{-1}(1 - \delta/q) \sqrt{h_i^{\mathrm{T}} \Sigma_z h_i} \leq b_i, \quad i = 1, \ldots, q. \tag{3.17}$$

Assuming a risk allocation *a priori* introduces conservatism into the problem formulation because the assumed allowed risk either might not be achieved or might be better allocated to another constraint.

The risk allocation method, in contrast, includes the allowed probability of violation for each univariate constraint as an optimization variable ϵ_i. In order to ensure that the total probability of violation is below the threshold, the optimization variables are restricted by requiring $\sum \epsilon_i \leq \delta$. For the application of Boole's method in this thesis, this risk allocation method was ultimately chosen as it provides increased system performance with only a slight increase in computational complexity.

Using this risk allocation technique and the evaluation of the chance constraints in Eqn. (3.15), if the constraints

$$\begin{gathered} 1 - \Phi\left(\frac{b_i - h_i^{\mathrm{T}}\bar{z}}{\sqrt{h_i^{\mathrm{T}}\Sigma_z h_i}}\right) \leq \epsilon_i, \quad i = 1, \ldots, q, \\ \textstyle\sum_{i=1}^{q} \epsilon_i \leq \delta \end{gathered} \tag{3.18}$$

are satisfied then the original joint chance constraint will also be satisfied. In [31], they showed that using this set of constraints makes the optimization program (2.10) a convex program if $\delta \leq 0.5$ due to the concavity of the Gaussian cumulative distribution function $\Phi(x)$ in the range $x \in [0, \infty)$.

Conservatism

Using Boole's inequality to upper bound the probability of constraint violation introduces conservatism into the solution. This section evaluates the impact of the conservatism on the quality of the solution.

Consider the following example with two random events A and B. The probability that either event A or B occurs is equal to $\mathrm{P}(A \bigcup B) = \mathrm{P}(A) + \mathrm{P}(B) - \mathrm{P}(A \bigcap B)$. Boole's inequality upper bounds the probability by $\mathrm{P}(A \bigcup B) \leq \mathrm{P}(A) + \mathrm{P}(B)$, resulting in a conservatism in this example of $\mathrm{P}(A \bigcap B)$ due to double counting the probability that both events occur, as shown in Figure 3.2.

To further investigate the conservatism of using Boole's inequality to bound the

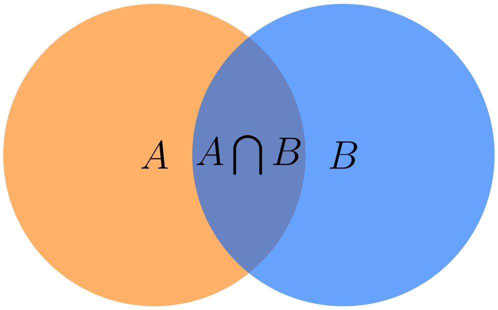


Figure 3.2: An illustration of the conservatism of Boole's inequality.

joint chance constraints, consider the following example with a set of random events $E_1, \ldots, E_m$. The events are assumed to be independent with a uniform probability of δ/m for any event occurring. The probability that at least one event occurs is

$$\begin{aligned} \mathrm{P}(E_1 \bigcup \ldots \bigcup E_m) &= 1 - \mathrm{P}(\bar{E}_1 \bigcap \ldots \bigcap \bar{E}_m) \\ &= 1 - \mathrm{P}(\bar{E}_1) \cdots \mathrm{P}(\bar{E}_m) \\ &= 1 - \left(1 - \frac{\delta}{m}\right)^m . \end{aligned} \tag{3.19}$$

Boole's inequality upper bounds the probability by $\mathrm{P}(E_1 \bigcup \ldots \bigcup E_m) \leq \delta$ resulting in the following conservatism

$$\delta - \left(1 - \left(1 - \frac{\delta}{m}\right)^m\right) . \tag{3.20}$$

For this example, Table 3.1 compares the conservatism in Boole's inequality for varying numbers of events and probability of the individual events occurring. For a fixed δ, as the number of events increases the conservatism is constant, however, as δ increases the conservatism also increases. For $m = 20$, the conservatism increases by over 400 times for $\delta = 0.001$ to $\delta = 0.5$. This occurs due to the increase in the probability of multiple events happening which accounts for the conservatism in Boole's inequality. Fortunately, for the problems that are of interest, the allowed probability of failure is typically less than 0.1 and therefore the conservatism in using Boole's inequality is small.

m	δ	Conservatism	Percentage
10	0.01	4.5e-05	0.45
20	0.01	4.7e-05	0.47
30	0.01	4.8e-05	0.48
40	0.01	4.9e-05	0.49
50	0.01	4.9e-05	0.49
20	0.001	4.7e-07	0.05
20	0.05	0.0012	2.34
20	0.1	0.0046	4.61
20	0.2	0.018	8.95
20	0.3	0.039	13.05
20	0.4	0.068	16.90
20	0.5	0.1	20.54

Table 3.1: A comparison of the conservatism in using Boole's inequality to bound the probability of failure for various number of constraints and allowed probability of constraint violation.

An illustration of using Boole's inequality for the joint chance constraint problem in Example 1 is shown in Figure 3.3. For this example, the risk allocation method is used. The red solid lines are the original constraint, the black line is Boole's approach, and the transformed constraints that result in the allowed probability of failure is shown as the blue solid line. As expected, for a small allowed probability of failure, the conservatism in using Boole's inequality to approximate the constraints is small. This can be seen as the true constraints and the constraints using Boole's inequality lie on top of one another.

3.2 Sampling Methods for Handling Chance Constraints

The difficulty in evaluating the chance constraints is in calculating the multivariate integral of the probability distribution for the constraints. The analytical methods presented in the last section assume a specific form of the probability distribution to simplify the evaluation. A different approach is presented in the following section that

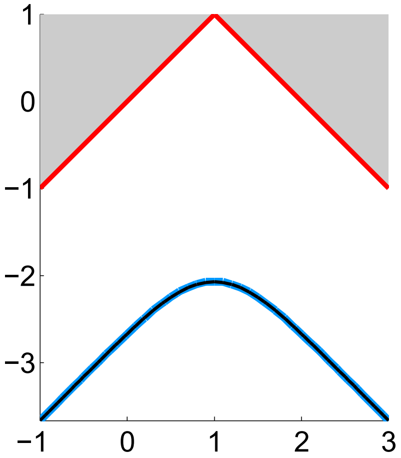


Figure 3.3: An illustration of using Boole's inequality to bound the joint chance constraint for Example 1. The original constraints are the red solid lines, the Boole's approach is shown as the black line, and the transformed constraints that result in the desired probability of failure are shown as the blue solid line. The blue and black lines are on top of each other. The feasible region for the state is below the corresponding lines.

uses a number of samples to represent the probability distribution which simplifies the evaluation of the chance constraints.

The following subsections use sampling to handle the chance constraints by using a finite set of particles to represent the probability distribution of the system; this converts the stochastic control problem into a deterministic one. Each of the following methods sample N_s particles at each time-step from the noise sources and initial state to obtain the sets

$$\begin{gathered}\{w_0^{(1)},\dots,w_{N-1}^{(1)},\dots,w_0^{(N_s)},\dots,w_{N-1}^{(N_s)}\},\\ \{v_0^{(1)},\dots,v_{N-1}^{(1)},\dots,v_0^{(N_s)},\dots,v_{N-1}^{(N_s)}\},\\ \{x_0^{(1)},\dots,x_0^{(N_s)}\}.\end{gathered} \tag{3.21}$$

An approximation of the distribution of the system state, measurement output, and control input can then be calculated using this set of samples:

$$\begin{gathered}x_{k+1}^{(j)} = A_k x_k^{(j)} + B_k u_k^{(j)} + W_k w_k^{(j)}\\ y_k^{(j)} = C_k y_k^{(j)} + V_k v_k^{(j)}\\ u_k^{(j)} = \varphi_k\left(y_0^{(j)},\dots,y_k^{(j)}\right)\end{gathered} \tag{3.22}$$

for all $j = 1, \ldots, N_s$. Let $\mathbb{X}^{(j)}$ be defined as $\mathbb{X}^{(j)} = \left[{x_0^{(j)}}^{\mathrm{T}}, \ldots, {x_N^{(j)}}^{\mathrm{T}} \right]^{\mathrm{T}}$ and similarly $\mathbb{U}^{(j)} = \left[{u_0^{(j)}}^{\mathrm{T}}, \ldots, {u_{N-1}^{(j)}}^{\mathrm{T}} \right]^{\mathrm{T}}$.

In the following subsections, three different approaches of using sampling to simplify the chance constraints are presented: mixed integer programming approach, scenario approach, and convex bounding approach.

3.2.1 Mixed Integer Programming

Several researchers [37, 38] proposed the idea of using mixed integer programming to approximate the chance constraints by counting the number of samples satisfying the constraint:

$$\mathrm{P}\left(H_i^{\mathrm{T}}\mathbb{X} + F_i^{\mathrm{T}}\mathbb{U} - b_i \leq 0\right) \approx \frac{1}{N_s}\left|\left\{H_i^{\mathrm{T}}\mathbb{X}^{(j)} + F_i^{\mathrm{T}}\mathbb{U}^{(j)} - b_i \leq 0, j = 1, \ldots, N_s\right\}\right| \tag{3.23}$$

for all $i = 1, \ldots, N_c$. In order to use this approximation to replace the chance constraints in the optimization program (2.10), a set of $N_c N_s$ binary variables $\{\beta_{11}, \ldots, \beta_{1N_s}, \ldots, \beta_{N_c 1}, \ldots, \beta_{N_c N_s}\}$ are introduced such that $\beta_{ij} \in \{0, 1\}$. The binary variables are defined such that $\beta_{ij} = 1$ implies that the constraint $H_i^{\mathrm{T}}\mathbb{X}^{(j)} + F_i^{\mathrm{T}}\mathbb{U}^{(j)} - b_i \leq 0$ holds for sample j. Consequently, the sum of the binary variables can be used to approximate the chance constraints

$$\frac{1}{N_s}\sum_{j=1}^{N_s}\beta_{ij} \geq 1 - \delta_i. \tag{3.24}$$

In order to use this technique, the restrictions on the binary variables, namely

$$\beta_{ij} = 1 \Rightarrow H_i^{\mathrm{T}}\mathbb{X}^{(j)} + F_i^{\mathrm{T}}\mathbb{U}^{(j)} - b_i \leq 0, \tag{3.25}$$

need to be imposed. One method of achieving this is by using the big-M method [75] with the binary variables resulting in the constraint

$$H_i^{\mathrm{T}}\mathbb{X}^{(j)} + F_i^{\mathrm{T}}\mathbb{U}^{(j)} - b_i \leq M\left(1 - \beta_{ij}\right), \tag{3.26}$$

where M is a large number such that if $\beta_{ij} = 0$ then the constraint will be satisfied for all $\mathbb{X}^{(j)}$ and $\mathbb{U}^{(j)}$. For completeness, the chance constraint in Eqn. (3.4) is replaced by the following two constraints

$$\begin{gathered} H_i^{\mathrm{T}} \mathbb{X}^{(j)} + F_i^{\mathrm{T}} \mathbb{U}^{(j)} - b_i \leq M \left(1 - \beta_{ij}\right), \ \forall i, j \\ \frac{1}{N_s} \sum_{j=1}^{N_s} \beta_{ij} \geq 1 - \delta_i, \ \forall i, j. \end{gathered} \tag{3.27}$$

There are several important properties of this method. First, neither an under- nor over-approximation can be guaranteed for the approximation of the chance constraints in Eqn. (3.23). Therefore this is only a heuristic method, which must be certified through cross validation. Second, as the number of samples tends to infinity, the approximation of the chance constraints will converge to the true probability distribution,

$$\frac{1}{N_s} \left| \left\{ H_i^{\mathrm{T}} \mathbb{X}^{(j)} + F_i^{\mathrm{T}} \mathbb{U}^{(j)} - b_i \leq 0, j = 1, \ldots, N_s \right\} \right|_{N_s \to \infty} \to \mathrm{P}\left(H_i^{\mathrm{T}} \mathbb{X} + F_i^{\mathrm{T}} \mathbb{U} - b_i \leq 0 \right). \tag{3.28}$$

While in theory this is an important aspect of this approach, in practice it is difficult to use a large number of particles due to the computational complexity. Third, using binary variables to keep track of whether or not the constraints have been violated drastically increases the computational complexity of solving the optimization program. Consequently, this approach can typically only use 100s of samples which might not be enough to sufficiently approximate the true probability distribution of the chance constraints.

A comparison of this approach for various numbers of samples is shown in Figure 3.4 for Example 1. The black dashed, green dash-dotted, and orange dotted lines show constraints with 100, 1000, and 5000 samples, respectively. The area below each line corresponds to the feasible region for the system that the technique estimates will satisfy the chance constraints. The blue solid line is the line that equals the desired probability of failure. As expected, the results vary drastically for the different number of samples and the best approximation is with the largest number of samples. As shown, the approximations can either be over- or under-approximations for the

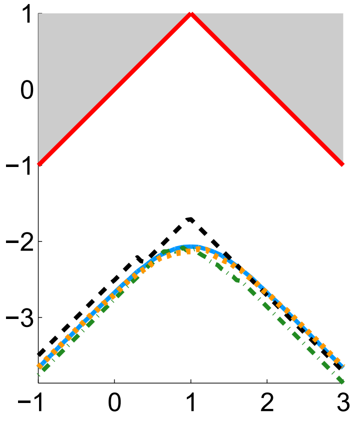


Figure 3.4: An illustration of the mixed integer programming technique for various number of samples for Example 1. The original constraints are the red solid lines, and the mixed integer programming technique is shown as the black dashed line, green dash-dotted line, and the orange dotted line for 100, 1000, and 5000 number of samples, respectively. The blue solid line is the line that equals the desired probability of failure. The feasible region for the state is below the corresponding lines.

original constraint and consequently there is no guarantee the chance constraint will be satisfied.

3.2.2 Scenario Approach

Another sampling technique, proposed by Calafiore and Campi [34] and De Fairas and Van Roy [35], is the scenario approach. Instead of using samples to approximate the probability distribution of the chance constraints, this method uses the samples to bound the chance constraints in order to guarantee they hold with probability $1 - \Delta$. Specifically, the scenario approach replaces the chance constraints in Eqn. (3.4) with the deterministic constraints

$$H_i^{\mathrm{T}} \mathbb{X}^{(j)} + F_i^{\mathrm{T}} \mathbb{U}^{(j)} - b_i \leq 0, \quad i = 1, \ldots, N_c,\ j = 1, \ldots, N_s. \tag{3.29}$$

In this approach, the number of samples required needs to be determined in order to guarantee that if Eqn. (3.29) is satisfied, then the original chance constraints will hold with probability $1 - \Delta$, or more formally,

$$\mathrm{P}\left(\mathrm{P}\left(H_i^{\mathrm{T}}\mathbb{X} + F_i^{\mathrm{T}}\mathbb{U} - b_i \leq 0\right) \geq 1 - \delta_i\right) \geq 1 - \Delta, \quad i = 1, \ldots, N_c. \tag{3.30}$$

Let $\delta \in (0, 1)$ be the smallest probability of failure for all the chance constraints $\delta = \min\{\delta_0, \ldots, \delta_{N_c}\}$, $\Delta \in (0, 1)$ be the confidence parameter for the scenario approach, and n be the number of optimization variables. Calafiore and Campi [34] showed that if

$$N_s \geq \left\lceil \frac{2}{\delta} \ln \frac{1}{\Delta} + 2n + \frac{2n}{\delta} \ln \frac{2}{\delta} \right\rceil$$

(where $\lceil \cdot \rceil$ denotes the smallest integer greater than or equal to the argument) then with probability $1 - \Delta$ the original chance constraints hold

$$\mathrm{P}\left(H_i^{\mathrm{T}}\mathbb{X} + F_i^{\mathrm{T}}\mathbb{U} - b_i \leq 0\right) \geq 1 - \delta_i, \quad i = 1, \ldots, N_c.$$

Intuitively, as the confidence that the original chance constraints hold increases, $\Delta \to 0$, the number of samples required goes to infinity. Consequently, the exponential growth of the required number of samples may be prohibitive for some problems. Figure 3.5 illustrates the growth of the number of required samples for $\delta = 0.03$ and $n = 2$.

An important property of the scenario approach is that since the joint chance constraints were replaced by a set of convex constraints, the solution to the transformed optimization program can be efficiently computed through numerical algorithms. This results in a dramatic speedup over the mixed integer programming method. In addition, with the mixed integer programming approach, the scenario approach formally guarantees the satisfaction of the chance constraints with probability $1 - \Delta$. However, this formal guarantee may come at the cost of an overly conservative solution or even result in infeasibility.

An example of the conservativeness of the solution is shown in Figure 3.6 for the problem in Example 1. The number of samples required for a $\Delta = 0.999$ was 1025. The original constraint is the red solid line and the scenario approach guarantees the satisfaction of the joint chance constraint with probability 0.999 by restricting the

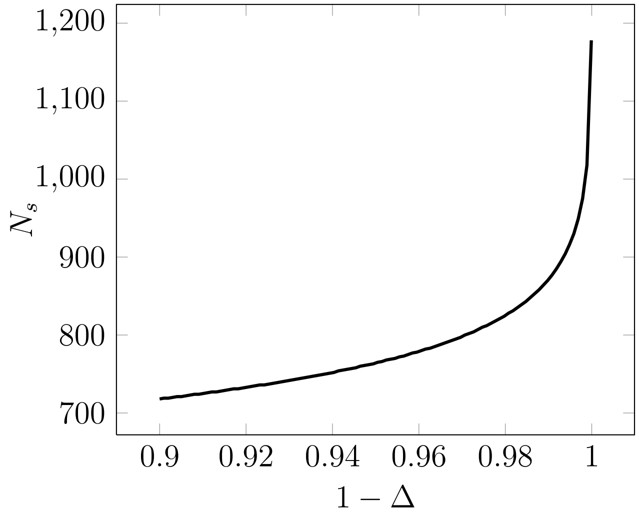


Figure 3.5: An illustration of the growth of the number of samples required as a function of Δ.

variable's feasible region to be below the black dotted line. The transformed constraints that result in the allowed probability of failure are shown as the blue solid lines. As is depicted in this example, the scenario approach results in a very conservative feasible region. In contrast to the mixed integer programming approach (or other Monte Carlo methods), the conservativeness (or performance) of the approach cannot be improved by using more samples. In fact, using more samples will only increase the conservativeness of the solution.

3.2.3 Convex Bounding Method

Another sampling approach is the convex bounding method developed by [76, 22]. Since the probability distribution of the chance constraints may not be a convex function, it is difficult to include them in the optimization program. This method finds a suitable conservative, convex approximation for the probability distribution of the chance constraints. The resulting problem is a convex program which may decrease the computational complexity.

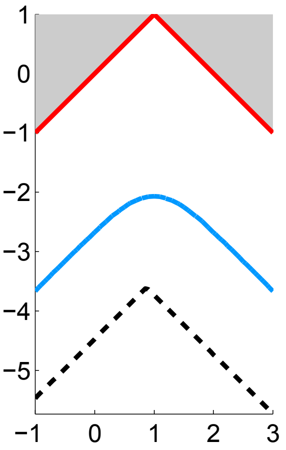


Figure 3.6: An illustration of the scenario approach for Example 1. The original constraints are the red solid lines, the scenario approach is shown as the black dotted line for 1025 number of samples, and the transformed constraints that result in the desired probability of failure are shown as the blue solid line. The feasible region for the state is below the corresponding lines.

Consider a single generic individual chance constraint of the form

$$\mathrm{P}\left(\psi\left(\mathbb{X},\mathbb{U}\right)\leq 0\right)\geq 1-\delta, \tag{3.31}$$

which is equivalent to

$$\mathrm{P}\left(\psi\left(\mathbb{X},\mathbb{U}\right)>0\right)\leq \delta. \tag{3.32}$$

The probability distribution in Eqn. (3.32) can be calculated via

$$\mathrm{P}\left(\psi\left(\mathbb{X},\mathbb{U}\right)>0\right)=\mathrm{E}\left[\mathbf{1}\left(\psi\left(\mathbb{X},\mathbb{U}\right)\right)\right] \tag{3.33}$$

where $\mathbf{1}(\cdot)$ is the indicator function defined as

$$\mathbf{1}(z)=\begin{cases}1, & \text{if } z>0\\ 0, & \text{otherwise.}\end{cases} \tag{3.34}$$

The indicator function $\mathbf{1}(z)$ (shown in Figure 3.7) is a nonconvex function, this

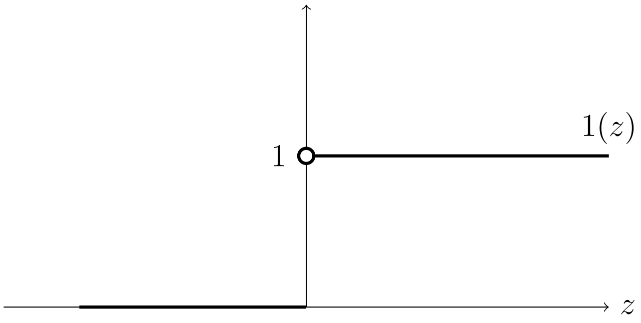


Figure 3.7: The indicator function $\mathbf{1}(z)$.

greatly complicates the evaluation of the chance constraints and solving the optimization problem (2.10). However, by bounding the indicator function by a convex function the optimization program simplifies to a convex program.

Suppose such a nonnegative, nondecreasing, convex function $\tilde{\psi} : \mathbb{R} \rightarrow \mathbb{R}$ can be found such that for any α, $\tilde{\psi}(z/\alpha) \geq \mathbf{1}(z)$ for all z then

$$\mathrm{E}\left[\tilde{\psi}\left(\psi\left(\mathbb{X},\mathbb{U}\right)/\alpha\right)\right] \geq \mathrm{P}\left(\psi\left(\mathbb{X},\mathbb{U}\right) > 0\right). \tag{3.35}$$

Consequently, if the following convex constraint is satisfied, then the original chance constraint in Eqn. (3.32) is guaranteed to hold:

$$\mathrm{E}\left[\tilde{\psi}\left(\psi\left(\mathbb{X},\mathbb{U}\right)/\alpha\right)\right] \leq \delta. \tag{3.36}$$

The constraint in Eqn. (3.36) holds for any α, but different choices of α may lead to a better approximation of the original chance constraint in Eqn. (3.32). Equation (3.36) can be rewritten as

$$\mathrm{E}\left[\alpha\tilde{\psi}\left(\psi\left(\mathbb{X},\mathbb{U}\right)/\alpha\right)\right] \leq \alpha\delta. \tag{3.37}$$

The lefthand side of Eqn. (3.37) is a perspective function $\alpha\tilde{\psi}\left(z/\alpha\right)$ that is convex in z and α for $\alpha > 0$. As shown in [11], the perspective of a convex function is also convex, therefore $\alpha\tilde{\psi}\left(\psi\left(\mathbb{X},\mathbb{U}\right)/\alpha\right)$ is convex in $\mathbb{X}$, $\mathbb{U}$ and α for $\alpha > 0$. Consequently, the constraint in Eqn. (3.37) is convex and one can optimize over α to obtain the best

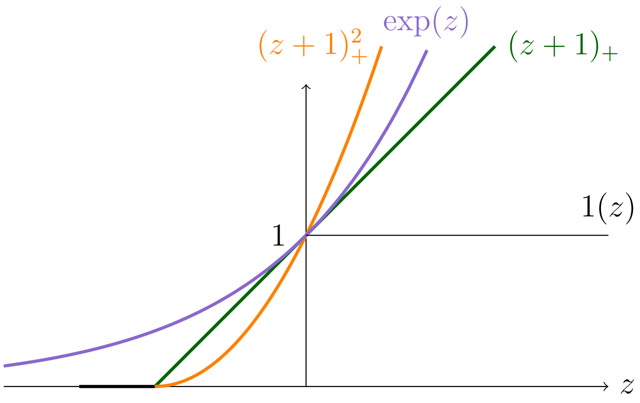


Figure 3.8: A comparison of several convex bounds that can be used to approximate the indicator function $\mathbf{1}(z)$. The functions $(z+1)_+$, $\exp(z)$, and $(z+1)_+^2$ are represented as the green, purple and orange lines respectively.

approximation, thereby reducing the conservativeness of the method.

$$\inf_{\alpha>0} \mathrm{E}\left[\alpha\tilde{\psi}\left(\psi\left(\mathbb{X},\mathbb{U}\right)/\alpha\right)\right] - \alpha\delta \leq 0 \tag{3.38}$$

Now that the form of the convex constraint used to bound the original chance constraint has been formulated, the next step is to determine what form of function to use for $\tilde{\psi}$. The restrictions on the function, as stated previously, are that it needs to be a convex function and $\tilde{\psi}(z) \geq \mathbf{1}(z)$ for all z. Several examples of possible functions are shown in Figure 3.8. The following subsections evaluate the bound in Eqn. (3.38) for each of these variants of $\tilde{\psi}(z)$.

Markov Bound (Conditional Value-at-Risk)

One possible functional form for $\tilde{\psi}(z)$ is $\tilde{\psi}(z) = (z+1)_+$ where the subscript $+$ denotes $\max\{z+1, 0\}$. This function results in the Markov bound for any $\alpha > 0$:

$$\begin{aligned} \mathrm{P}\left(\psi\left(\mathbb{X}, \mathbb{U}\right) > 0\right) &= \mathrm{P}\left(\psi\left(\mathbb{X}, \mathbb{U}\right) + \alpha > \alpha\right) \\ &= \mathrm{P}\left(\left(\psi\left(\mathbb{X}, \mathbb{U}\right) + \alpha\right)_+ > \alpha\right) \\ &< \mathrm{E}\left[\frac{\left|\left(\psi\left(\mathbb{X}, \mathbb{U}\right) + \alpha\right)_+\right|}{\alpha}\right] \\ &= \mathrm{E}\left[\left(\psi\left(\mathbb{X}, \mathbb{U}\right)/\alpha + 1\right)_+\right]. \end{aligned} \tag{3.39}$$

The convex constraint that bounds the original chance constraint is then:

$$\mathrm{E}\left[\alpha\left(\psi\left(\mathbb{X}, \mathbb{U}\right)/\alpha + 1\right)_+\right] \le \alpha\delta \tag{3.40}$$

which can be rewritten as

$$\mathrm{E}\left[\left(\psi\left(\mathbb{X}, \mathbb{U}\right) + \alpha\right)_+\right] \le \alpha\delta. \tag{3.41}$$

When this bound is used, it is often referred to as the conditional value-at-risk (CVaR) bound, first introduced in [76]. To show the equivalence, rewrite Eqn. (3.41) as

$$\frac{\mathrm{E}\left[\left(\psi\left(\mathbb{X}, \mathbb{U}\right) + \alpha\right)_+\right]}{\delta} - \alpha \le 0. \tag{3.42}$$

Since the lefthand side is convex in α, the minimum over α can be constructed

$$\inf_{\alpha > 0}\left(\frac{\mathrm{E}\left[\left(\psi\left(\mathbb{X}, \mathbb{U}\right) + \alpha\right)_+\right]}{\delta} - \alpha\right), \tag{3.43}$$

which is also convex in $\mathbb{X}$ and $\mathbb{U}$. This results in the well known quantity CVaR.

Chebyshev Bound

Another functional form used for $\tilde{\psi}(z)$ is $\tilde{\psi}(z) = (z+1)_+^2$ which yields the Chebyshev bound,

$$\mathrm{P}\left(\psi\left(\mathbb{X},\mathbb{U}\right)>0\right) \leq \mathrm{E}\left[\left(\psi\left(\mathbb{X},\mathbb{U}\right)/\alpha+1\right)_+^2\right]. \tag{3.44}$$

Using this function, the convex constraint that bounds the original chance constraint is

$$\mathrm{E}\left[\alpha\left(\psi\left(\mathbb{X},\mathbb{U}\right)/\alpha+1\right)_+^2\right] \leq \alpha\delta \tag{3.45}$$

which can also be rewritten as

$$\mathrm{E}\left[\left(\psi\left(\mathbb{X},\mathbb{U}\right)+\alpha\right)_+^2/\alpha\right] \leq \alpha\delta. \tag{3.46}$$

Traditional Chebyshev Bound

If the subscript + is dropped off the previous function, then a more conservative bound is obtained

$$\mathrm{E}\left[\alpha\left(\psi\left(\mathbb{X},\mathbb{U}\right)/\alpha+1\right)^2\right] \leq \alpha\delta, \tag{3.47}$$

and after expanding terms and simplifying it can be rewritten as

$$2\mathrm{E}\left[\psi\left(\mathbb{X},\mathbb{U}\right)\right]+\frac{1}{\alpha}\mathrm{E}\left[\psi\left(\mathbb{X},\mathbb{U}\right)^2\right]+\alpha(1-\delta) \leq 0. \tag{3.48}$$

Minimizing the previous equation over α gives $\alpha = \left(\frac{1}{1-\delta}\mathrm{E}\left[\psi\left(\mathbb{X},\mathbb{U}\right)^2\right]\right)^{1/2}$ which yields the final constraint

$$\mathrm{E}\left[\psi\left(\mathbb{X},\mathbb{U}\right)\right]+\left(\left(1-\delta\right)\mathrm{E}\left[\psi\left(\mathbb{X},\mathbb{U}\right)^2\right]\right)^{1/2} \leq 0. \tag{3.49}$$

Notably, this final constraint only depends upon the first and second moments of the function $\psi\left(\mathbb{X},\mathbb{U}\right)$. This property could be beneficial if the first and second moments can be easily computed from the probability distributions of the uncertainty in the system, allowing the constraints to be transformed into deterministic constraints.

Chernoff Bound

Another function that could be used is $\tilde{\psi}(z) = \exp(z)$ which leads to the Chernoff bound. The use of this function was also called the "Bernstein approximation" by Nemirovski and Shapiro [22]. This leads to the constraint for any $\alpha > 0$:

$$\mathrm{P}\left(\psi\left(\mathbb{X}, \mathbb{U}\right) > 0\right) \leq \mathrm{E}\left[\exp\left(\psi\left(\mathbb{X}, \mathbb{U}\right)/\alpha\right)\right], \tag{3.50}$$

with the corresponding convex bound for the original chance constraint given by:

$$\mathrm{E}\left[\alpha \exp\left(\psi\left(\mathbb{X}, \mathbb{U}\right)/\alpha\right)\right] \leq \alpha\delta. \tag{3.51}$$

A comparison of these different convex bounding techniques is shown in Figure 3.9. The constraint considered is $\mathrm{P}(b - x \leq 0) \geq 0.9$ where $b \sim \mathcal{N}(0, 1)$. The probability distribution of b is plotted as well as the solutions for x using each of the previous functions used to bound the indicator function. The vertical lines from left to right are the exact solution (red), Markov bound $(z+1)_+$ (green), Chebyshev bound $(z+1)_+^2$ (orange), Chernoff bound $\exp(z)$ (purple), and traditional Chebyshev bound $(z+1)^2$ (brown). The feasible region for each solution is to the right of the line. The least conservative solution uses the Markov bound with function $(z+1)_+$, followed closely by the Chebyshev bound with function $(z+1)_+^2$. The most conservative solution uses the traditional Chebyshev bound with function $(z+1)^2$.

Reducing the Conservatism of the Markov Bound Approach

Hong et al. [23] devised a scheme to reduce the conservatism in the Markov bound (CVaR) approach by providing a better approximation of the indicator function. This is illustrated in Figure 3.10. Figure 3.10(a) displays the Markov bound approximation, $\tilde{\psi}(z)$, of the indicator function and a right shifted version of it as $\hat{\psi}(z)$. If the difference of these two convex functions is used to bound the indicator function, it will result in a better approximation as shown in Figure 3.10(b).

Using the difference of the functions $\tilde{\psi}(z) = (z+1)_+$ and $\hat{\psi}(z) = (z)_+$ to bound

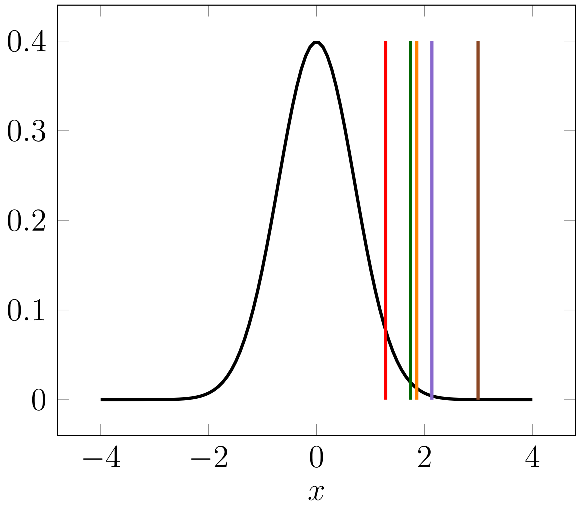


Figure 3.9: A comparison of the solutions using the different bounding functions of the indicator function. The vertical lines from left to right are the exact solution (red), Markov bound $(z+1)_+$ (green), Chebyshev bound $(z+1)_+^2$ (orange), Chernoff bound $\exp(z)$ (purple), and traditional Chebyshev bound $(z+1)^2$ (brown). The feasible region for each solution is to the right of the line. The black curve is the probability distribution of the constraint.

the probability of constraint violation yields

$$\mathrm{P}\left(\psi\left(\mathbb{X},\mathbb{U}\right)>0\right)\leq \mathrm{E}\left[\left(\psi\left(\mathbb{X},\mathbb{U}\right)+\alpha\right)_+\right]-\mathrm{E}\left[\left(\psi\left(\mathbb{X},\mathbb{U}\right)\right)_+\right]. \tag{3.52}$$

The best upper bound of the chance constraints can be obtained by optimizing over α. If this is used, then it can be shown that it equals the probability of constraint violation i.e.

$$\begin{aligned}\inf_{\alpha>0}\mathrm{E}\left[\left(\psi\left(\mathbb{X},\mathbb{U}\right)+\alpha\right)_+\right]-\mathrm{E}\left[\left(\psi\left(\mathbb{X},\mathbb{U}\right)\right)_+\right]&=\lim_{\alpha\to 0}\mathrm{E}\left[\left(\psi\left(\mathbb{X},\mathbb{U}\right)+\alpha\right)_+\right]-\mathrm{E}\left[\left(\psi\left(\mathbb{X},\mathbb{U}\right)\right)_+\right]\\&=\mathrm{P}\left(\psi\left(\mathbb{X},\mathbb{U}\right)>0\right).\end{aligned} \tag{3.53}$$

Consequently, the following constraint is used to replace the original chance constraint

$$\inf_{\alpha>0}\mathrm{E}\left[\left(\psi\left(\mathbb{X},\mathbb{U}\right)+\alpha\right)_+\right]-\alpha\delta-\mathrm{E}\left[\left(\psi\left(\mathbb{X},\mathbb{U}\right)\right)_+\right]\leq 0. \tag{3.54}$$

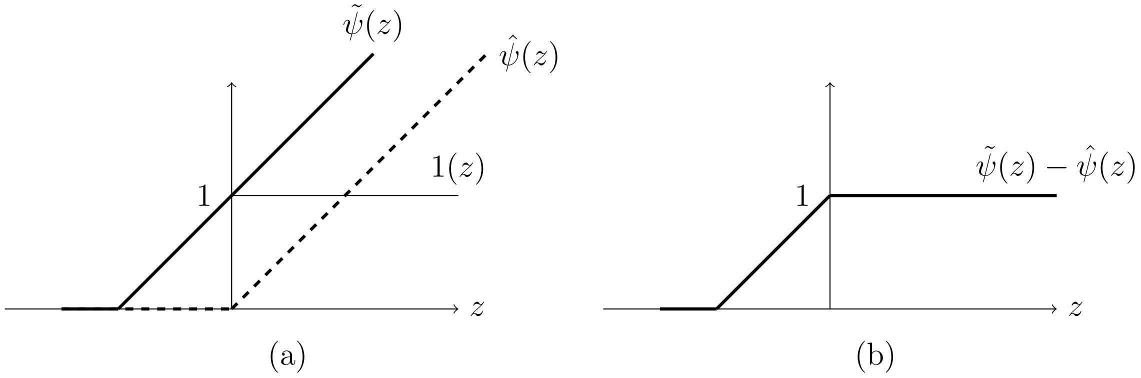


Figure 3.10: A difference of two convex functions used to approximate the indicator function $\mathbf{1}(z)$. (a) $\tilde{\psi}(z)$ is the function used to bound the indicator function in the Markov bound method and $\hat{\psi}(z)$ is a right shifted version of the function. (b) The approximation of the indicator function using the difference of two convex functions $\tilde{\psi}(z)$ and $\hat{\psi}(z)$ results in a tighter bound of the indicator function.

The first two terms in Eqn. (3.54) correspond to the Markov bound (CVaR) constraint and therefore the conservatism in the Markov bound approach is equal to $\mathrm{E}\left[(\psi\left(\mathbb{X},\mathbb{U}\right))_{+}\right]$. Unfortunately Eqn. (3.54) in general is a nonconvex function, however, the optimization program can be solved through sequential convex approximations by using a first order Taylor series approximation of the third term around the previous solution. It was shown in [23] that this will converge to a Karush-Kuhn-Tucker point of the original chance constrained problem.

While this approach provides a better solution than the Markov bound approach, generally it comes at the cost of increased computational complexity because it needs to solve this sequence of convex approximations.

Handling the Joint Chance Constraints

The convex bounding approaches were introduced above using individual chance constraints, but they can also be used with joint chance constraints. The joint chance constraint can be rewritten as

$$\mathrm{P}\left(\psi_j\left(\mathbb{X},\mathbb{U}\right) \leq 0,\ j = 1,\ldots,q\right) = \mathrm{P}\left(\max\{\psi_1\left(\mathbb{X},\mathbb{U}\right),\ldots,\psi_q\left(\mathbb{X},\mathbb{U}\right)\} \leq 0\right). \tag{3.55}$$

Let ψ be defined as $\psi\left(\mathbb{X},\mathbb{U}\right) = \max\{\psi_1\left(\mathbb{X},\mathbb{U}\right),\ldots,\psi_q\left(\mathbb{X},\mathbb{U}\right)\}$, which can be used in the previous convex bounding approaches. Since $\psi_j\left(\mathbb{X},\mathbb{U}\right),\ j = 1,\ldots,q$ are convex functions, $\psi\left(\mathbb{X},\mathbb{U}\right)$ is a convex function and therefore the convexity of the previous bounding approaches will be conserved.

Solving the Convex Bounding Method

The difficulty in using the convex bounding approaches is in evaluating the expectation in Eqn (3.37). Typically, this expectation is not trivial to solve analytically since it involves multidimensional integrals, however, it can be easily evaluated through sampling. The main issue when using sampling is the solution obtained may not satisfy the original chance constraints as observed with the other sampling methods. However, since the convex bounding approach results in a convex optimization program, which typically can be solved efficiently, a large number of samples can be used to better approximate the probability distributions. Cross-validation can also be used to estimate the confidence that the original chance constraints are satisfied for the solution.

A comparison of the Markov and Chebyshev bounding functions, shown as the blue dashed and orange dash-dotted lines, for the previous joint chance constraint problem in Example 1 is shown in Figure 3.11. As expected, this method results in a conservative prediction of the feasible region for the problem, and the result from the Chebyshev bound is more conservative than the Markov bound.

Interpretation of the Convex Bounding Methods

By using the indicator function in Eqn. (3.33), the original chance constraints equally weight all violations of the constraints no matter how large. In contrast, the convex bounding methods use functions that approximate the indicator function and weight larger constraint violations more than smaller ones. This can be visualized in Figure 3.8.

A good example highlighting the difference between what the original chance constraints and the convex bounding functions measure is given in [77]. Consider

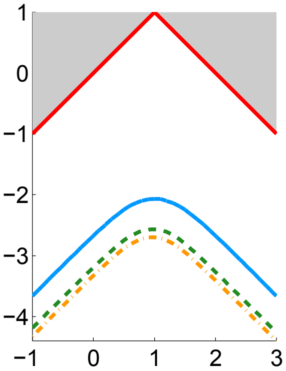


Figure 3.11: An illustration of the convex bounding approach for Example 1. The Markov and Chebyshev functions shown as the blue dashed and orange dash-dotted lines, respectively. The original constraints are the red solid lines, and the transformed constraints that result in the desired probability of failure are shown as the blue solid line. The feasible region for the state is below the corresponding lines.

the following example in portfolio optimization contrasting the differences between a trader and the owner of the investment company. A trader may prefer high uncontrolled risks because the incentive structure rewards large returns. In the worst case, if the trader does realize a huge loss, the trader may be fired but they may move to another company. Consequently, the trader may perceive a loss of $100 dollars to be the same as $100,000 dollars. In contrast, the company owner would have to cover the large losses if they are realized. Therefore, the owner of the company would weight a $100,000 dollars loss much higher than the $100 dollars loss.

The "correct" choice of whether to weight constraint violations equally or not comes down to the specific application. In stochastic motion planning for aerial vehicles through uncertain environments, it doesn't matter whether it is estimated the aerial vehicle will crash into a wall by a violation of 1 cm or 1 m because the aerial vehicle will most likely not be able to recover no matter what the constraint violation is. In contrast, how much the constraints are violated plays a crucial role in stochastic motion planning for steerable needles in the human body. In this application, the constraints are critical regions such as organs that the needle shouldn't pass through and the potential for damage increases as the needle penetrates these regions more.

3.3 Comparison of Analytic and Sampling Methods

There are many different approaches to handle the joint chance constraints, but there is no clear-cut correct choice in all situations. Table 3.2 compares the approaches presented in this chapter in computational complexity, conservativeness of the approach, and whether it can handle arbitrary distributions. The ellipsoidal relaxation technique has the smallest computational complexity, but achieves this through increasing the conservativeness of the solution. By using Boole's inequality, the conservativeness is drastically reduced from the ellipsoidal relaxation method while the computational complexity only increases slightly. Also, the original chance constraints are only guaranteed to hold for certain distributions. The mixed integer programming approach has a very large computational complexity which limits its use for only small sized problems. Although it can handle arbitrary distributions, the mixed integer programming approach approximates the constraints and therefore the original chance constraints are not guaranteed to hold. The scenario approach results in a convex optimization program with moderate computational complexity. While it can handle arbitrary distributions, the method bounds the original chance constraints resulting in a very conservative solution which in most situations will limit its usefulness. The final approach is to use the convex bounding approach CVaR in-place of the original chance constraints. Since CVaR is a bound for the original chance constraint it will guarantee the feasibility of the original problem. This approach has moderate computational complexity compared to the other approaches, and can handle arbitrary distributions through sampling but results in a conservative solution.

A comparison of the different approaches is shown in Figure 3.12 for Example 1. The methods mixed integer programming (MIP), Boole's inequality, conditional value-at-risk, ellipsoidal relaxation, and scenario approach are shown as the purple dotted line, solid black line, orange dotted line, green dash-dotted line, and brown solid line, respectively. The line corresponding to the allowed probability of failure is shown as the blue solid line which is below the solid black line for the Boole's inequality method. As stated before, the mixed integer programming method is an

	Complexity	Conservativeness	Arbitrary distributions
Ellipsoidal	very low	large	approx.
Boole's	low	small	approx.
MIP	very large	—	yes
Scenario	medium	very large	yes
Convex bound	medium	medium	yes

Table 3.2: Comparison of the different approaches of handling the original chance constraints.

approximation method and consequently violates the desired probability of failure in some regions while being conservative in others. The other methods ranked in terms of conservativeness are: Boole's inequality, CVaR, ellipsoidal relaxation, and scenario approach. As shown, the scenario approach is very conservative as compared to the other approaches. Also note that Boole's inequality and CVaR are able to take into account the interaction of the two constraints at the cusp, located at $x = 1$, resulting in a curved feasible region. The other approaches disregard this interaction resulting in a $\wedge$-shaped feasible region.

While there might not be a correct choice for the method that is used to handle the joint chance constraints, two of them are not realistic to use. The MIP approach has a very large computational cost, resulting in only being usable for small sized problems. Unfortunately, the problems that are of concern in this thesis are too large to be solved efficiently with the MIP method. The scenario approach can also be disregarded due to the very large conservatism introduced into the problem formulation, often resulting in an infeasible optimization program.

The other three methods (ellipsoidal relaxation method, CVaR method, and Boole's method) have their own trade-offs resulting in application dependent choices. The ellipsoidal relaxation method transforms the stochastic optimization program into a linear program for fixed controllers which is the least computationally intensive, however, it is the most conservative and is only an approximate solution for non-Gaussian distributions. Boole's method provides the least conservative solution with only a slight increase in computational complexity, but is only an approximate solution for non-Gaussian distributions. The CVaR method resorts to sampling to

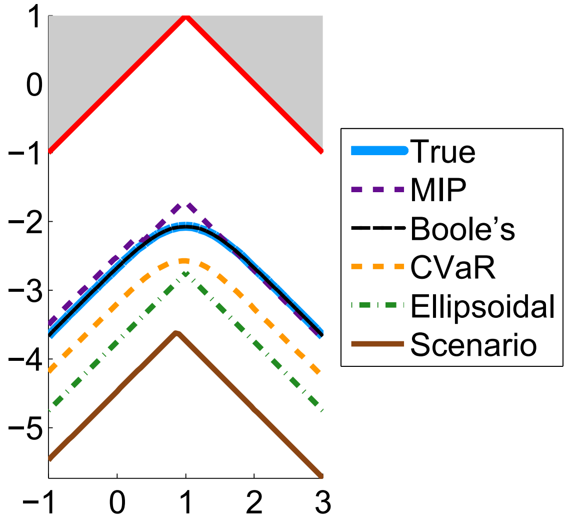


Figure 3.12: A comparison of the mixed integer programming (MIP), Boole's inequality, conditional value-at-risk (CVaR), ellipsoidal relaxation, and scenario approach methods for handling the joint chance constraints shown as the purple dotted line, solid black line, orange dotted line, green dash-dotted line, and brown solid line, respectively. The true probability of failure line is shown as the solid blue line which is directly under the line for Boole's inequality. The predicted feasible region for each method is below the corresponding lines. The original constraints are the red solid lines.

handle arbitrary probability distributions, but has the largest computational complexity. The conservativeness of the CVaR method is between the Boole's and ellipsoidal relaxation methods.

3.4 Discussion on Constraint Modeling

The previous sections compared the conservatism of the different approaches for handling the joint chance constraints. In this section, the mathematical properties of two different risk measures are explored.

In the original problem formulation, a set of constraints were imposed on the system

$$\psi_i\left(\mathbb{X}, \mathbb{U}\right) \leq 0, \quad i = 1, \ldots, N_c. \tag{3.56}$$

However, due to the uncertainty in the system the constraints are not guaranteed to always hold. Therefore some measure of the risk associated with violating the constraints needs to be used. In risk management, there are many different ways to measure the risk, but typically there are two that are commonly used: Value-at-Risk (VaR) and conditional value-at-risk.

Given a random variable $z \in \mathbb{R}$ and confidence level $1-\delta$, value-at-risk is defined as

$$\mathrm{VaR}\left(z; 1-\delta\right) = \inf\left\{\gamma \mid \mathrm{P}\left(z \leq \gamma\right) \geq 1-\delta\right\}. \tag{3.57}$$

The chance constraint $\mathrm{P}\left(z \leq 0\right) \geq 1-\delta$ is the same as $\mathrm{VaR}\left(z; 1-\delta\right) \leq 0$. Conditional value-at-risk is defined as

$$\mathrm{CVaR}\left(z; 1-\delta\right) = \inf_{\beta}\left(\beta + \frac{1}{1-(1-\delta)}\mathrm{E}\left[z-\beta\right]_{+}\right). \tag{3.58}$$

Let β^* be defined as $\beta^* = \mathrm{VaR}\left(z; 1-\delta\right)$ such that $\mathrm{P}\left(z \geq \beta^*\right) = \delta$, then CVaR can also be written for continuous distributions as

$$\begin{aligned}
\mathrm{E}\left[z|z \geq \beta^*\right] &= \mathrm{E}\left[\beta^* + \left(z-\beta^*\right)|z \geq \beta^*\right] \\
&= \beta^* + \frac{\mathrm{E}\left[z-\beta^*\right]_{+}}{\mathrm{P}\left(z \geq \beta^*\right)} \\
&= \mathrm{CVaR}\left(z; 1-\delta\right).
\end{aligned} \tag{3.59}$$

This expression explains where the name conditional value-at-risk comes from because it is the expected value conditioned on the value-at-risk.

There has been a large amount of controversy over whether VaR or CVaR is a better measure of risk [78, 79, 77, 80, 81, 82]. Antzer et al. [78] and Rockafellar [79] compared the two approaches axiomatically by determining a set of properties that a good measure of risk should possess. Let ρ be defined as an arbitrary measure of risk, X be a random variable, $\|X\|_2 = \left(\mathrm{E}\left[X^2\right]\right)^{1/2}$, and X^k be sequence of random variables. If $\|X^k - X\|_2 \to 0$ as $k \to \infty$ then X^k, $k = 1, 2, \ldots$ converges to a random variable X with respect to this norm. Rockafellar [79] defined a risk measure ρ to be *coherent* if the following axioms hold:

1. $\rho\left(C\right) = C$ for all constant C;

2. $\rho(\lambda X_1 + (1-\lambda)X_2) \leq \lambda\rho(X_1) + (1-\lambda)\rho(X_2)$;
3. $\rho(X_1) \leq \rho(X_2)$ if $X_2 \leq X_1$;
4. $\rho(X) \leq 0$ when $\|X^k - X\|_2 \to 0$ with $\rho\left(X^k\right) \leq 0$;
5. $\rho(\lambda X) = \lambda\rho(X)$ for $\lambda > 0$.

The combination of (2) and (5) leads to the *subadditivity* property

$$\rho(X_1 + X_2) \leq \rho(X_1) + \rho(X_2). \tag{3.60}$$

Using this definition of coherency it can be shown that VaR is not a coherent measure of risk for all distributions because it violates the subadditivity property. The following example illustrates this concept.

Example 2 ([82])**.** *Consider two scalar random variables X_1 and X_2 that are independent and given by the following distribution*

$$X_i = \epsilon_i + \eta_i,\ \epsilon_i \sim \mathcal{N}(0,1),\ \eta_i = \begin{cases} 0 & \textit{with probability}\, 0.991 \\ -10 & \textit{with probability}\, 0.009 \end{cases},\ \forall i = 1,2.$$

In this example $\mathrm{VaR}(X_i; 0.99) = 3.1$ *since the probability of* $\eta = -10$ *is less than* 0.01*, and* $\mathrm{VaR}(X_1 + X_2; 0.99) = 9.8$ *because the probability of getting the* -10 *draw for either* X_i *is larger than* 0.01*. Consequently,*

$$\mathrm{VaR}(X_1 + X_2; 0.99) = 9.8 > \mathrm{VaR}(X_1; 0.99) + \mathrm{VaR}(X_2; 0.99) = 6.2.$$

In financial risk management the subadditivity property plays a key role because it ensures that the diversification principle of portfolio theory holds. In essence, this means that a diversified portfolio has a lower risk than a non-diversified portfolio; if this were not the case, then it would be beneficial for someone to open multiple accounts to make their investments instead of having one central account.

While VaR is not coherent for general probability distributions, the property does hold for elliptical distributions, which includes the normal distribution [83]. In [82], the authors showed that for fat-tailed distributions VaR is subadditive in the tail regions.

The previous discussion on constraint modeling is not advocating one approach over another but rather presenting the material for a complete characterization of the advantages and disadvantages of the approaches.

Chapter 4

Stochastic Control with Deterministic Constraint Parameters

In Chapter 2 the stochastic control problem was formally defined, but in general it is intractable to solve. Chapter 3 described several different methods of relaxing the problem to obtain a tractable problem formulation. Of the six different methods presented, only three approaches (Boole's method, the CVaR method and the ellipsoidal relaxation method) are realistic to use in bounding (or approximating) the original problem. The remainder of this thesis will demonstrate the use of these three approaches in the proposed framework to solve the following stochastic control problems: stochastic control with deterministic constraint parameters, stochastic control with feedback design, predictive control of jump Markov systems, stochastic control with uncertain constraint parameters, and dual control for uncertain constraint parameters. The thesis concludes with a real-time experimental demonstration of the proposed framework applied to stochastic motion planning for a quadrotor vehicle navigating through an uncertain environment.

This chapter develops the stochastic control framework with deterministic parameters using a fixed controller and estimator. In this development, the properties of the closed-loop system are directly incorporated into evaluating the joint chance constraints. This evaluation is performed using Boole's method, the CVaR method, and the ellipsoidal relaxation method. The solutions of each method are compared through several examples at the end of the chapter. As compared to traditional deterministic methods, the presented framework directly handles the inherit stochasticity without drastically increasing the computational complexity.

The presented examples include application of the developed framework to the stochastic motion planning problem through known environments. Traditionally, deterministic motion planners are used to solve this problem, however, the system would almost surely experience some form of disturbances, leading to failures in the system. Such disturbances may include wind, wheel slippage, or unmodeled dynamics. Also, the system would not be able to sense its state exactly, which would also introduce uncertainty into the system. Deterministic planners typically account for the system's uncertainty by adding buffers to the constraints, however, these buffers are usually chosen in an ad-hoc manner and may not provide the level of safety the user desires. Instead, by using the proposed stochastic control framework, the probability of constraint violation over the entire trajectory can be successfully constrained to be less than a user allowed amount.

4.1 System Description

For the stochastic control problem in this chapter, the initial state, process noise and measurement noise are assumed to be independent, Gaussian random variables distributed according to:

$$x_0 \sim \mathcal{N}(\bar{x}_0, \Sigma_0), \quad w_k \sim \mathcal{N}(0, \Sigma_w) \quad \forall k, \quad v_k \sim \mathcal{N}(0, \Sigma_v) \quad \forall k. \tag{4.1}$$

Only the system trajectory is constrained probabilistically (rather than state and control), leading to the joint chance constraint $\mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} - b \leq 0\right) \geq 1 - \delta$ (or equivalently $\mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} - b > 0\right) \leq \delta$). The means of the control inputs are constrained to lie in a region $\bar{\mathbb{U}} \in F_{\mathbb{U}}$, where $F_{\mathbb{U}}$ is a convex, feasible region for the control inputs. In this chapter, a Kalman filter [84] is chosen as the estimator and the system is controlled via a linear quadratic trajectory tracking controller, as these are the most widely used estimator and controller pair in industry. The following subsections review the formulation of this particular estimator and controller choice. The presented formulation, however, is applicable to other estimator and controller choices.

4.1.1 Kalman Filter

For a linear, Gaussian system, the Kalman filter is the minimum mean square error estimator for the mean and uncertainty of the system state. Let $\hat{\Sigma}_{k|k-1}$ be the covariance matrix of the optimal estimate of x_k given the measurements $\{y_0, \ldots, y_{k-1}\}$, and let $\hat{\Sigma}_{k|k}$ be the covariance matrix of the optimal estimate of x_k given the measurements $\{y_0, \ldots, y_k\}$. The Kalman filter operates through the following recursion. The measurement update is

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + L_k \left(y_k - C_k \hat{x}_{k|k-1} \right) \tag{4.2}$$

$$\hat{\Sigma}_{k|k} = \left(I - L_k C_k \right) \hat{\Sigma}_{k|k-1} \tag{4.3}$$

where $I \in \mathbb{R}^{n \times n}$ is the identity matrix and

$$L_k = \hat{\Sigma}_{k|k-1} C_k^{\mathrm{T}} \left(C_k \hat{\Sigma}_{k|k-1} C_k^{\mathrm{T}} + \Sigma_v \right)^{-1}, \tag{4.4}$$

for all $k = 0, \ldots, N$. The process update is

$$\hat{x}_{k|k-1} = A_{k-1} \hat{x}_{k-1|k-1} + B_{k-1} u_{k-1}, \tag{4.5}$$

$$\hat{\Sigma}_{k|k-1} = A_{k-1} \hat{\Sigma}_{k-1|k-1} A_{k-1}^{\mathrm{T}} + \Sigma_w, \tag{4.6}$$

for all $k = 1, \ldots, N$. The Kalman filter recursion starts from $\hat{x}_{0|-1} = \bar{x}_0$ and $\hat{\Sigma}_{0|-1} = \Sigma_0$.

One of the key properties of the Kalman filter is that the covariance, $\hat{\Sigma}_{k|k}$, can be pre-computed *before* any measurements are received which allows it to be utilized in the stochastic control process.

4.1.2 Linear Quadratic Trajectory Control

Given perfect state information, the linear quadratic controller minimizes the following cost function,

$$J = \underset{u_k,\, k=0,\ldots,N-1}{\text{minimize}} \sum_{\tau=0}^{N} \left(x_\tau - x_\tau^d\right)^{\mathrm{T}} Q \left(x_\tau - x_\tau^d\right) + \sum_{\tau=0}^{N-1} u_\tau^{\mathrm{T}} R u_\tau, \tag{4.7}$$

where x_k^d is the desired trajectory at time-step k, $Q \in \mathbb{R}^{n\times n}$ is the state error cost, and $R \in \mathbb{R}^{m\times m}$ is the control input cost. It can be shown that the optimal input for a linear system is an affine function of the current state given by,

$$u_k^* = K_k x_k + g_k \tag{4.8}$$

$$K_k = -\left(R + B_k^{\mathrm{T}} P_{k+1} B_k\right)^{-1} B_k^{\mathrm{T}} P_{k+1} A_k \tag{4.9}$$

$$g_k = -\left(R + B_k^{\mathrm{T}} P_{k+1} B_k\right)^{-1} B_k^{\mathrm{T}} q_{k+1} \tag{4.10}$$

where P_k and q_k are calculated via,

$$P_k = Q + A_k^{\mathrm{T}} P_{k+1} A_k - A_k^{\mathrm{T}} P_{k+1} B_k \left(R + B_k^{\mathrm{T}} P_{k+1} B_k\right)^{-1} B_k^{\mathrm{T}} P_{k+1} A_k \tag{4.11}$$

$$q_k = \left(A_k + B_k K_k\right)^{\mathrm{T}} q_{k+1} - Q x_k^d \tag{4.12}$$

starting from $P_N = Q$ and $q_N = -Q^{\mathrm{T}} x_N^d$.

4.1.3 Linear Quadratic Gaussian

Given that there is process and measurement noise, the exact value of the state x_k is not known preventing direct evaluation of the cost function and controller in Eqns. (4.7), (4.8). Rather, only the expectation of the cost function in Eqn. (4.7) can be minimized. Fortunately, the certainty equivalence principle [85] has been used to show that the optimal controller for a linear, Gaussian system with quadratic cost uses the estimate of the state from a Kalman filter in Eqn. (4.8), i.e.,

$$u_k^* = K_k \hat{x}_{k|k} + g_k. \tag{4.13}$$

4.2 *A Priori* Distribution of the Closed-Loop State and Control Input

Now that the estimator and controller have been defined, the next step in defining the stochastic control problem is to formulate the probability distribution of the closed-loop system state, $\mathbb{X}$, and closed-loop control input, $\mathbb{U}$. These will then be used to evaluate the joint chance constraints.

Applying linear system theory, the closed-loop uncertainty of the system state can be characterized *a priori* before any measurements are received. Given the assumption of the Kalman filter estimator and linear quadratic trajectory tracking controller, the true system evolves according to,

$$x_{k+1} = A_k x_k + B_k(K_k \hat{x}_{k|k} + g_k) + W_k w_k, \quad k = 0, \ldots, N-1, \tag{4.14}$$

and the estimate of the system state evolves according to,

$$\hat{x}_{k+1|k+1} = A_k \hat{x}_{k|k} + B_k \left(K_k \hat{x}_{k|k} + g_k\right) + L_{k+1} \left(y_{k+1} - C_{k+1} \hat{x}_{k+1|k}\right), \quad k = 0, \ldots, N-1. \tag{4.15}$$

Substituting in for y_{k+1} and $\hat{x}_{k+1|k}$ from Eqns. (2.4) and (4.5) into Eqn. (4.15) yields,

$$\begin{aligned} \hat{x}_{k+1|k+1} = \; & A_k \hat{x}_{k|k} + B_k \left(K_k \hat{x}_{k|k} + g_k\right) + \\ & L_{k+1} \left[C_{k+1} \left(A_k x_k + B_k(K_k \hat{x}_{k|k} + g_k) + W_k w_k\right) + \right. \\ & \left. V_{k+1} v_{k+1} - C_{k+1} \left(A_k \hat{x}_{k|k} + B_k \left(K_k \hat{x}_{k|k} + g_k\right)\right)\right]. \end{aligned} \tag{4.16}$$

Finally, simplifying and collecting the common terms results in

$$\begin{aligned} \hat{x}_{k+1|k+1} = \; & \left(A_k + B_k K_k - L_{k+1} C_{k+1} A_k\right) \hat{x}_{k|k} + L_{k+1} C_{k+1} A_k x_k + B_k g_k + \\ & L_{k+1} C_{k+1} W_k w_k + L_{k+1} V_{k+1} v_{k+1}. \end{aligned} \tag{4.17}$$

The state estimation error, $e_{k|k} = x_k - \hat{x}_{k|k}$, then evolves according to

$$e_{k+1|k+1} = (I - L_{k+1}C_{k+1})A_k e_{k|k} + (I - L_{k+1}C_{k+1})W_k w_k - L_{k+1}V_{k+1}v_{k+1}. \tag{4.18}$$

To determine the distribution of the closed-loop system, x_{k+1} and $e_{k+1|k+1}$ from Eqns. (4.14) and (4.18) are combined into a single system resulting in the following time-varying system,

$$z_{k+1} = F_k z_k + \bar{B} g_k + G_k s_k \tag{4.19}$$

where

$$\begin{aligned} z_k &= \begin{bmatrix} x_k \\ e_{k|k} \end{bmatrix}, \quad s_k = \begin{bmatrix} w_k \\ v_{k+1} \end{bmatrix} \quad \bar{B} = \begin{bmatrix} B_k \\ 0 \end{bmatrix}, \\ F_k &= \begin{bmatrix} A_k + B_k K_k & -B_k K_k \\ 0 & (I - L_{k+1}C_{k+1})A_k \end{bmatrix}, \\ G_k &= \begin{bmatrix} W_k & 0 \\ (I - L_{k+1}C_{k+1})W_k & -L_{k+1}V_{k+1} \end{bmatrix} \end{aligned} \tag{4.20}$$

with $s_k \sim \mathcal{N}(0, \Sigma_s)$, $\Sigma_s = \operatorname{diag}(\Sigma_w, \Sigma_v)$ and diag places the matrices along the diagonal.

The system starts from $z_0 = \begin{bmatrix} x_0^{\mathrm{T}}, & x_0^{\mathrm{T}} - \hat{x}_{0|0}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. The mean, $\bar{z}_k$, and covariance, $\tilde{\Sigma}_k$, for the system defined by Eqn. (4.19) can now be determined,

$$\bar{z}_{k+1} = F_k \bar{z}_k + \bar{B} g_k \tag{4.21}$$

$$\tilde{\Sigma}_{k+1} = F_k \tilde{\Sigma}_k F_k^{\mathrm{T}} + G_k \Sigma_s G_k^{\mathrm{T}}, \tag{4.22}$$

starting from $\bar{z}_0 = \begin{bmatrix} \hat{x}_{0|0} \\ 0 \end{bmatrix}$ and $\tilde{\Sigma}_0 = \begin{bmatrix} \Sigma_0 & 0 \\ 0 & \hat{\Sigma}_{0|0} \end{bmatrix}$. Finally, the closed-loop state and control input evolves according to,

$$\begin{bmatrix} x_k \\ u_k \end{bmatrix} = \underbrace{\begin{bmatrix} I & 0 \\ K_k & -K_k \end{bmatrix}}_{\Lambda_k} \begin{bmatrix} x_k \\ e_{k|k} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} g_k. \tag{4.23}$$

Consequently, the *a priori* closed-loop state and control input at time-step k, for all

$k = 1, \ldots, N-1$, is distributed by,

$$\begin{bmatrix} x_k \\ u_k \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} T_k^{xx_0}\bar{x}_0 + T_k^{xg}\mathbf{g} \\ K_k(T_k^{xx_0}\bar{x}_0 + T_k^{xg}\mathbf{g}) + g_k \end{bmatrix}, \Lambda_k \tilde{\Sigma}_k \Lambda_k^{\mathrm{T}} \right) \tag{4.24}$$

where $\mathbf{g} = \begin{bmatrix} g_0^{\mathrm{T}} & \ldots & g_{N-1}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. The k-th row of the matrix T^{xx_0} is,

$$T_k^{xx_0} = (A_{k-1} + B_{k-1}K_{k-1})(A_{k-2} + B_{k-2}K_{k-2}) \cdots (A_0 + B_0 K_0) \tag{4.25}$$

and the elements of the lower triangular matrix T^{xg} are,

$$T^{xg} = \begin{bmatrix} B_0 & 0 & \ldots & 0 \\ (A_1 + B_1K_1)B_0 & B_1 & \ldots & 0 \\ (A_2 + B_2K_2)(A_1 + BK_1)B_0 & (A_2 + BK_2)B_1 & B_2 & 0 \\ & & \vdots & \\ (\prod_{\tau=1}^{N-1}(A_\tau + B_\tau K_\tau))B_0 & (\prod_{\tau=2}^{N-1}(A_\tau + B_\tau K_\tau))B_1 & \ldots & B_{N-1} \end{bmatrix}. \tag{4.26}$$

Let $\mathbb{X}^d$ and $\mathbf{q}$ be defined as: $\mathbb{X}^d = \begin{bmatrix} x_1^{d\mathrm{T}} & \ldots & x_N^{d\ \mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ and $\mathbf{q} = \begin{bmatrix} q_1^{\mathrm{T}} & \ldots & q_N^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. The constant input term in the LQG controller is defined as a function of $\mathbf{q}$ (recall from Eqn. (4.10)), i.e., $\mathbf{g} = \Psi\ \mathbf{q}$ where Ψ is a block diagonal matrix with entries $\Psi_{ii} = -(R + B_i^{\mathrm{T}} P_{i+1} B_i)^{-1} B_i^{\mathrm{T}}$, for all $i = 0, \ldots, N-1$. Also, the vector $\mathbf{q}$ from Eqn. (4.12) can be written in terms of the desired trajectory, $\mathbf{q} = \Theta \mathbb{X}^d$ where Θ is defined as $\forall i, j \in [0, N-1]$,

$$\Theta_{ij} = \begin{cases} 0 & \text{if } j < i \\ -Q^{\mathrm{T}} & \text{if } j = i \\ -(\prod_{\tau=i+1}^{j}(A_\tau + B_\tau K_\tau)^{\mathrm{T}})Q^{\mathrm{T}} & \text{otherwise.} \end{cases} \tag{4.27}$$

Consequently, the mean of the closed-loop state is

$$\bar{\mathbb{X}} = T^{xx_0}\bar{x}_0 + T^{xx^d}\mathbb{X}^d \tag{4.28}$$

where $T^{xx^d} = T^{xg}\Psi\Theta$. The mean of the control inputs is

$$\bar{\mathbb{U}} = T^{ux_0}\bar{x}_0 + T^{ux^d}\mathbb{X}^d \tag{4.29}$$

with $T^{ux_0} = \tilde{K}T^{xx_0} + \begin{bmatrix} K_0 \\ 0 \end{bmatrix}$, $T^{ux^d} = \left(\tilde{K}T^{xx^d} + \Psi\Theta\right)$, and $\tilde{K}$ is a block matrix defined as

$$\tilde{K} = \begin{bmatrix} 0 & \dots & 0 \\ K_1 & & \vdots \\ 0 & \ddots & 0 \\ & & K_{N-1} \end{bmatrix}. \tag{4.30}$$

4.3 Gradient and Hessian of the Chance Constraints

To aid in the solution of the optimization program (2.10) using Boole's method, the gradient, g_c, and Hessian, H_c, of the chance constraints can also be computed analytically. First, the gradient can be computed for one probability constraint and then generalized for all. Let h_i be the i-th column of the constraint matrix H. From the chain rule,

$$\nabla_{\mathbb{X}^d}\mathrm{P}\left(h_i^{\mathrm{T}}\mathbb{X} > b_i\right) = \frac{\partial\mathrm{P}\left(h_i^{\mathrm{T}}\mathbb{X} > b_i\right)}{\partial\bar{\mathbb{X}}}\nabla_{\mathbb{X}^d}\bar{\mathbb{X}}. \tag{4.31}$$

From Eqn. (3.15) and the Leibniz integral rule,

$$\begin{aligned} \frac{\partial\mathrm{P}\left(h_i^{\mathrm{T}}\mathbb{X} > b_i\right)}{\partial\bar{\mathbb{X}}} &= -\frac{\partial\Phi\left(\dfrac{b_i - h_i^{\mathrm{T}}\bar{\mathbb{X}}}{\sqrt{h_i^{\mathrm{T}}\tilde{\Sigma}h_i}}\right)}{\partial\bar{\mathbb{X}}} \\ &= \frac{1}{\sqrt{2\pi h_i^{\mathrm{T}}\tilde{\Sigma}h_i}}\exp\left(-\frac{(b_i - h_i^{\mathrm{T}}\bar{\mathbb{X}})^2}{2h_i^{\mathrm{T}}\tilde{\Sigma}h_i}\right)h_i^{\mathrm{T}}. \end{aligned} \tag{4.32}$$

The gradient of $\bar{\mathbb{X}}$ with respect to $\mathbb{X}^d$ is $\nabla_{\mathbb{X}^d}\bar{\mathbb{X}} = T^{xx^d}$. Combining the individual gradients into vector notation yields the total constraint gradient:

$$g_c = T^{xx^d\,\mathrm{T}}Hl \tag{4.33}$$

where l is defined as a vector with elements,

$$l_i = \frac{1}{\sqrt{2\pi h_i^{\mathrm{T}} \tilde{\Sigma} h_i}} \exp\left(-\frac{(b_i - h_i^{\mathrm{T}} \bar{\mathbb{X}})^2}{2 h_i^{\mathrm{T}} \tilde{\Sigma} h_i}\right). \tag{4.34}$$

The Hessian of the probability constraints is determined through a similar process. The Hessian of one individual constraint is,

$$\nabla^2_{\mathbb{X}^d} \mathrm{P}\left(h_i^{\mathrm{T}} \mathbb{X} > b_i\right) = d_i {T^{xx^d}}^{\mathrm{T}} h_i h_i^{\mathrm{T}} T^{xx^d} \tag{4.35}$$

where $d_i = \dfrac{b_i - h_i^{\mathrm{T}} \bar{\mathbb{X}}}{\sqrt{2\pi} (h_i^{\mathrm{T}} \tilde{\Sigma} h_i)^{3/2}} \exp\left(-\dfrac{(b_i - h_i^{\mathrm{T}} \bar{\mathbb{X}})^2}{2 h_i^{\mathrm{T}} \tilde{\Sigma} h_i}\right)$. The Hessian of all the probability constraints in matrix form is then,

$$H_c = {T^{xx^d}}^{\mathrm{T}} H \mathrm{diag}\left(d\right) H^{\mathrm{T}} T^{xx^d}. \tag{4.36}$$

4.4 Final Optimization Program

The final optimization program for the stochastic control problem is shown below.

$$\begin{aligned} \underset{\mathbb{X}^d}{\text{minimize}} \quad & \mathbf{E}\left[\phi(\mathbb{X}, \mathbb{U})\right] \\ \text{subject to} \quad & \\ & \bar{\mathbb{X}} = T^{xx_0} \bar{x}_0 + T^{xx^d} \mathbb{X}^d \\ & \bar{\mathbb{U}} = T^{ux_0} \bar{x}_0 + T^{ux^d} \mathbb{X}^d \\ & \bar{\mathbb{U}} \in F_{\mathbb{U}} \\ & \text{joint chance constraints handling} \end{aligned} \tag{4.37}$$

The difference between each of the methods (Boole's, CVaR, and ellipsoidal relaxation) lies in how the joint chance constraints are handled. For Boole's method the following constraints are added

$$\begin{gathered} 1 - \Phi\left(\frac{b_i - h_i^{\mathrm{T}}\bar{\mathbb{X}}}{\sqrt{h_i^{\mathrm{T}}\Sigma_{\mathbb{X}}h_i}}\right) \leq \epsilon_i, \quad i = 1, \ldots, N_c \\ \textstyle\sum_{i=1}^{N_c} \epsilon_i \leq \delta. \end{gathered} \tag{4.38}$$

For the CVaR method the constraints are

$$\mathrm{E}\left[\left(\max(H^{\mathrm{T}}\mathbb{X} - b + \alpha)\right)_+\right] \leq \alpha\delta, \tag{4.39}$$

where the expectation is evaluated using sampling. Finally, the constraints for the ellipsoidal relaxation method are

$$h_i^{\mathrm{T}}\bar{\mathbb{X}} + r\sqrt{h_i^{\mathrm{T}}\Sigma_{\mathbb{X}}h_i} \leq b_i, \quad i = 1, \ldots, N_c, \tag{4.40}$$

where r is defined in Eqn. (3.9). For all of the methods, the resulting optimization program is convex.

4.5 Simulation Examples

The following section presents several examples comparing the solution to a stochastic control problem using Boole's method, the CVaR method, and the ellipsoidal relaxation method.

4.5.1 Open-Loop Unstable Dynamics, Convex Environment

This first example considers the case of an open-loop unstable system which could not be handled in [31] because they only considered the open-loop system distribution. By incorporating a feedback controller, the formulation presented in this chapter is shown to handle such unstable systems as long as a linear quadratic feedback law can stabilize the system.

Example 3. *Consider an unstable system with dynamics given by,*

$$A = \begin{bmatrix} 2.72 & 0 \\ 0.17 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0.17 \\ 7.2e-3 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

and a time horizon of $N = 20$. The noise parameters and the distribution of the initial state are,

$$\Sigma_w = 0.0001I, \quad \Sigma_v = 0.0001I, \quad x_0 \ \sim \ \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, 0.0001I\right).$$

The feasible region for the system state is defined by the following constraints,

$$\begin{bmatrix} 1 & 0 \end{bmatrix} x_k \leq 1.05, \quad \begin{bmatrix} 0 & -1 \end{bmatrix} x_k \leq 0.2, \quad \begin{bmatrix} -1 & 1 \end{bmatrix} x_k \leq 0.3$$

for all $k = 1, \ldots, N$. The allowed probability of constraint violation is $\delta = 0.01$. The objective function for this problem is quadratic in the final state as well as the control inputs,

$$\phi(\mathbb{X}, \mathbb{U}) = \left(x_N - x_{\text{ref}}\right)^T Q_{\text{obj}} \left(x_N - x_{\text{ref}}\right) + \mathbb{U}^T R_{\text{obj}} \mathbb{U} \tag{4.41}$$

with $Q_{\text{obj}} = I$, $R_{\text{obj}} = 0.001I$ and $x_{\text{ref}} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$.

The solutions for this example using Boole's method, the CVaR method, and the ellipsoidal relaxation method are shown in Figure 4.1(a)-(c), respectively. The start and goal location are shown by the blue circle and red 'x', the feasible region is the white area, the mean system trajectory is the blue, thick line, and the ellipses represent the 99.7% confidence ellipsoids at each time-step. The number of particles used in the CVaR method is $N_s = 1000$ and the solution for every particle is shown as the gray lines in Figure 4.1(b).

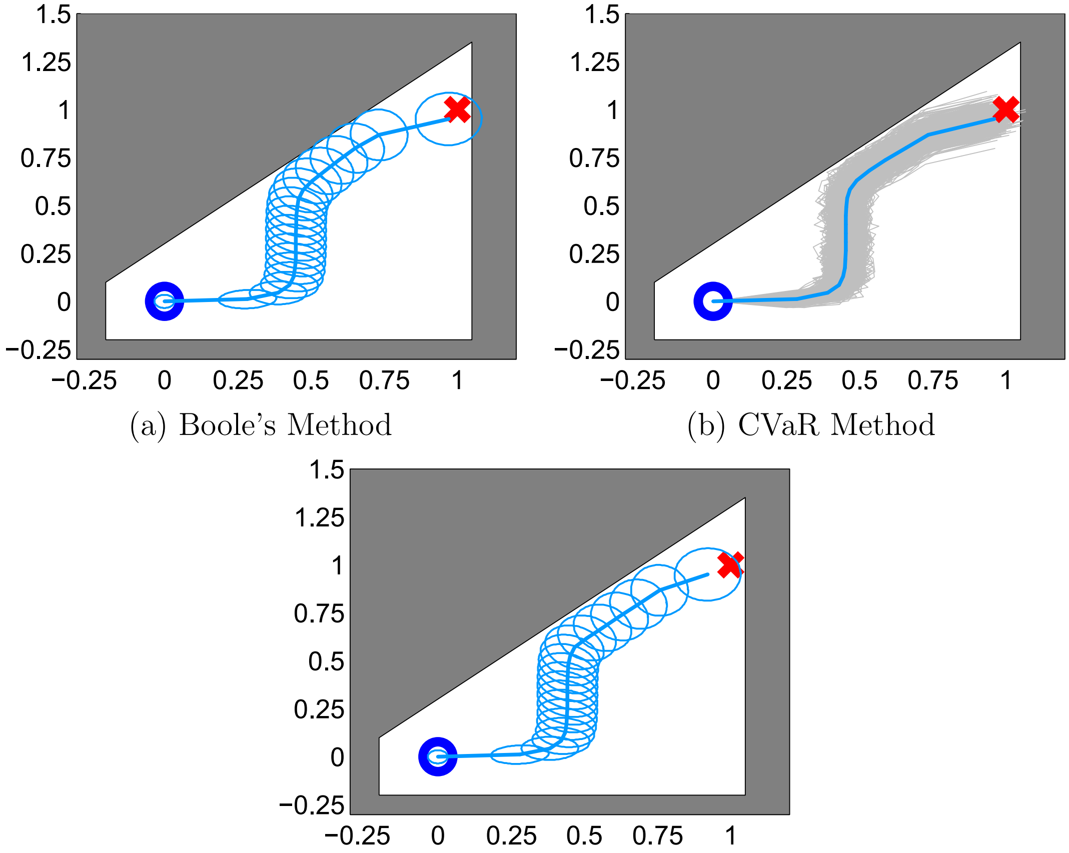
Table 4.1 compares the actual probability of constraint violation calculated through Monte Carlo simulation and the computation time for each of the methods. The optimization was performed using Matlab on a 2.7 GHz Intel Core i7 computer; the computation times are provided only for comparison and they can be drastically reduced by using a compiled language such as C++. Note, the computation time for the CVaR method is slightly inflated due to the inefficiency of loops in Matlab. As expected, using Boole's method results in the least conservative solution and the ellipsoidal relaxation method results in the most conservative solution. This can be seen visually in Figure 4.1 by how far the solution for each method stays away from the slanted boundary. Specifically, for Boole's method the uncertainty ellipses touch the slanted constraint, whereas for the ellipsoidal relaxation method the uncertainty ellipses are further away.

Table 4.1: A comparison of the calculated probability of constraint violation through Monte Carlo simulation for the methods. The allowed probability of failure is $\delta = 0.01$.

	δ_{sim}	comp. time (s)
Boole's method	0.0096	2.42
CVaR method ($N_s = 1000$)	0.0058	22.72
Ellipsoidal relaxation method	0.00031	0.84

Figure 4.2 displays the constraint violation probability at each time-step using Boole's method for the slanted line. At the start of the trajectory, the system has a very low chance of violating the constraints, but near the end the probability of violation increases dramatically. This non-uniform distribution of constraint violation probability is a result of using the risk allocation algorithm, which in this case has chosen to assign most of the allowable constraint violation near the end of the trajectory. This can be seen visually by the proximity of the system trajectory to the constraint in Figure 4.1(a). By using this risk allocation algorithm, the performance of the system is increased as compared with using a pre-defined risk profile.

A comparison of the probability of constraint violation statistics for the CVaR method with varying number of particles is shown in Table 4.2. The statistics are calculated for 100 runs of each algorithm and the probability of constraint violation is calculated via Monte Carlo simulation for each iteration. As expected, for the smaller number of particles the actual probability of constraint violation is larger due to the inability to represent the true probability distribution with smaller sample sizes. In some cases, the true probability of constraint violation exceeds the allowed limit. Consequently, the solution using the CVaR method must be checked using



(a) Boole's Method

(b) CVaR Method

(c) Ellipsoidal Relaxation Method

Figure 4.1: The solution for the unstable dynamics example for Boole's method, the CVaR method and the ellipsoidal relaxation method for Example 3. The blue circle is the starting position of the state and the red 'x' is the goal location. The white area is the feasible region. The thick blue line in each plot is the solution for the mean trajectory of the system. (a) The solution using Boole's method. The ellipses are the 99.7% confidence regions of the state at each time-step. (b) The solution using the CVaR method with 1000 particles. The trajectories for all the particles are shown as the gray lines. (c) The solution using the ellipsoidal relaxation method.

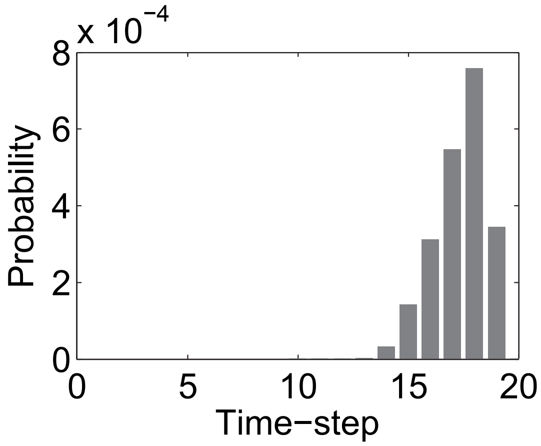


Figure 4.2: The probability of violating the slanted line constraint for the open-loop unstable example for Example 3. The horizontal axis is the time-step and the vertical axis is the probability of violating the constraint.

cross-validation to ensure that the original constraints hold. Also, as the number of particles increases to better represent the true distribution, the solution becomes more conservative.

Table 4.2: A comparison of the solution statistics for the CVaR algorithm for various number of particles.

N_s	mean δ_{sim}	std. dev. δ_{sim}	max δ_{sim}
500	0.0090	0.0037	0.0197
1000	0.0060	0.0022	0.0146
2000	0.0047	0.0015	0.0093

4.5.2 Nonconvex Environments with Integrator Dynamics

In the previous example, the environment was readily describable by a set of convex constraints. In general, however, the feasible region for stochastic control problems may be nonconvex regions, significantly increasing the computational complexity of solving the resulting optimization program. To handle these nonconvex cases, typically, the optimization program is transformed into a series of convex problems by either performing branch and bound on the disjunction of the constraints [86] or decomposing the feasible region into convex tunnels which are then planned through [87]. The latter approach is used in the following examples.

Example 4. *Consider a system with double integrator dynamics and a 2D position,*

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0.5\Delta t^2 & 0 \\ 0 & 0.5\Delta t^2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{4.42}$$

where $\Delta t = 0.1$ seconds and a time-horizon of $N = 20$. The noise parameters are,

$$\begin{gathered} \Sigma_w = \text{diag}(0.0003, 0.0005, 0.0003, 0.0005), \\ \Sigma_v = \text{diag}(0.001, 0.002). \end{gathered} \tag{4.43}$$

The objective function is the same as in Example 3 with $x_{\text{ref}} = \begin{bmatrix} 2 & 1 & 0 & 0 \end{bmatrix}^T$, $Q_{\text{obj}} = \begin{bmatrix} 10I & 0 \\ 0 & 0 \end{bmatrix}$, and $R_{\text{obj}} = 0.001I$.

Using the tunnel method to deal with the nonconvex environments, the feasible region can be decomposed into two different tunnels corresponding to going around the top or bottom of the first obstacle. Figure 4.3 shows the decomposition of the nonconvex region for the top and bottom paths.

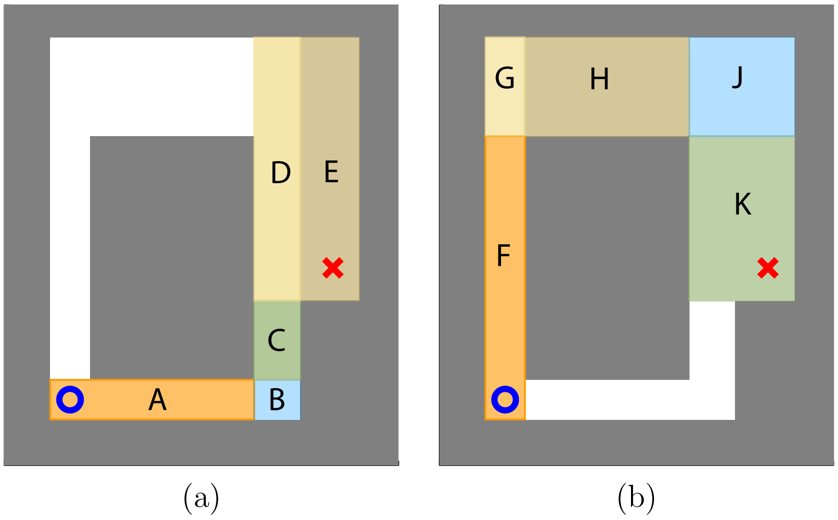


Figure 4.3: The decomposition of the nonconvex space into tunnels corresponding to the bottom path and top path in (a) and (b), respectively.

Bottom Region

The probability constraints for the bottom region are,

$$(A,B)\begin{bmatrix}-1 & 0 & 0 & 0\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & -1 & 0 & 0\end{bmatrix} x_k \le \begin{bmatrix}0.15\\ 1.75\\ 0.15\\ 0.15\end{bmatrix} \qquad k=1,\dots,k_1$$

$$(B,C,D)\begin{bmatrix}-1 & 0 & 0 & 0\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & -1 & 0 & 0\end{bmatrix} x_k \le \begin{bmatrix}-1.4\\ 1.75\\ 2.75\\ 0.15\end{bmatrix} \qquad k=k_1+1,\dots,k_2$$

$$(D,E)\begin{bmatrix}-1 & 0 & 0 & 0\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & -1 & 0 & 0\end{bmatrix} x_k \le \begin{bmatrix}-1.4\\ 1.75\\ 2.75\\ -0.75\end{bmatrix} \qquad k=k_2+1,\dots,N$$

where k_1 and k_2 are the switching times between the three sets of constraints. The switching times correspond to when the system exits one region, defined by the constraints, and enters another. Corresponding to the regions in Figure 4.3(a), the first inequality constrains the system to be in regions A and B, the second inequality restricts the system to only be in regions B, C and D, and the final constraint allows the system to be in regions D and E.

Top Region

Similarly, the probability constraints for the top region are,

$$
\begin{array}{rcccl}
(F,G) \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} x_k & \leq & \begin{bmatrix} 0.15 \\ 0.15 \\ 2.75 \\ 0.15 \end{bmatrix} & & k = 1, \ldots, k_1 \\
(G,H,J) \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} x_k & \leq & \begin{bmatrix} 0.15 \\ 1.75 \\ 2.75 \\ -2 \end{bmatrix} & & k = k_1 + 1, \ldots, k_2 \\
(J,K) \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} x_k & \leq & \begin{bmatrix} -1.4 \\ 1.75 \\ 2.75 \\ -0.75 \end{bmatrix} & & k = k_2 + 1, \ldots, N
\end{array}
$$

where again k_1 and k_2 are the switching times between the three sets of constraints. In reference to the regions in Figure 4.3(b), the first inequality constrains the system to be in regions F and G, the second inequality restricts the system to only be in regions G, H and J, and the final constraint allows the system to be in regions J and K.

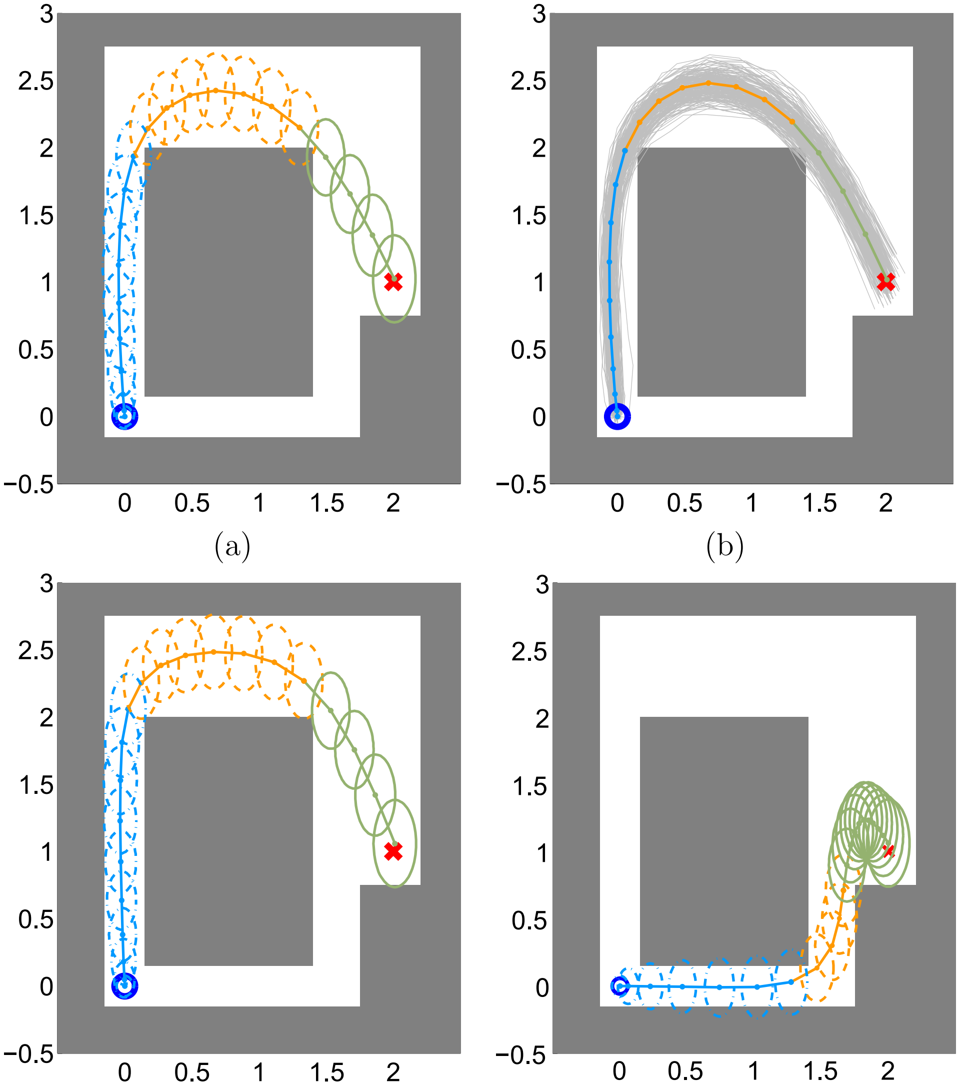
Results

For this example, the optimal path is always through the top region even though a route through the bottom region is shorter. This is due to the large uncertainty of the vertical position of the state which renders the bottom region infeasible with respect to the chance constraints for choices of $\delta < 0.19$. Figure 4.4(a)-(c) displays the optimal solution resulting from using Boole's method, the CVaR method, and the ellipsoidal relaxation method for $\delta = 0.1$. The optimal switching times for each algorithm are $k_1 = 9$ and $k_2 = 16$. The computed constraint violation from Monte Carlo simulation for Boole's method was 0.095 (note the conservativeness due to Boole's inequality). The number of particles chosen for the CVaR method was 200 and the calculated probability of constraint violation is 0.042.

Qualitatively, both methods produce similar solutions, however, Boole's method takes more risk resulting in a 5% lower objective function cost. For all solutions, the optimal trajectory curves toward the infeasible region in the beginning instead of following the center of the corridor. This initial deviation improves the overall objective and is allowed because the rest of the path has a small probability of violating the constraints.

A comparison of the probability of constraint violation statistics for different numbers of particles in the CVaR method is shown in Table 4.3. For each number of particles, 100 iterations of the algorithm was performed and the probability of constraint violation was calculated through a Monte Carlo simulation. It is interesting to note that as compared to the Example 3, a smaller number of particles is required to have the maximum probability of constraint violation be below the allowed probability.

As stated above, the optimal route is around the top of the obstacle because the bottom route is infeasible. Figure 4.4(d) shows a suboptimal solution, using Boole's method, for a path through the bottom region for an allowed constraint violation of 0.19. The system quickly follows the center of the corridor in the beginning, then slowly approaches the goal location at the end in order to reduce the probability of violating any constraints before the end of the planning horizon.



(a)

(b)

(c)

(d)

Figure 4.4: The results for planning through the nonconvex region in Example 4. The blue line with dash-dot ellipses, orange line with dash ellipses, and green line with solid ellipses indicate which three sets of constraints are active. (a) The solution using Boole's method. (b) The solution using the CVaR method with $N_s = 200$. The gray lines are the trajectories for each of the particles. (c) The solution using the ellipsoidal relaxation method. (d) The results for planning through the bottom route of the nonconvex region using Boole's method with an allowed constraint violation of $\delta = 0.19$.

Table 4.3: A comparison of the solution statistics for the CVaR algorithm for various number of particles for a nonconvex known environment.

N_s	mean δ_{sim}	std. dev. δ_{sim}	max δ_{sim}
100	0.061	0.021	0.151
200	0.047	0.015	0.091
300	0.042	0.010	0.068
400	0.042	0.0088	0.069
500	0.039	0.0070	0.062

4.5.3 Collision Avoidance

For the final example, a dynamic nonconvex motion planning example is chosen in which multiple vehicles have to avoid each other while attempting to reach their own goal locations. In contrast to the previous examples, this problem has a larger state space and includes state coupling in the joint chance constraints. In the following scenario, the vehicles were required to keep a minimum separation distance of 0.25 meters, which by definition is a nonconvex feasible region. Consequently, the same approach will be employed as in the previous section to simplify the problem into convex subproblems.

Example 5. *Consider a collision avoidance scenario with three vehicles. All vehicles have the same double integrator dynamics described by Eqn.* (4.42) *with* $\Delta t = 0.1$ *seconds, a time-horizon of* $N = 20$ *and noise parameters given in Eqn.* (4.43)*.*

The objective function is,

$$\phi(\mathbb{X}, \mathbb{U}) = \sum_{i=1}^{3} \left(x_N^i - x_{\text{ref}}^i\right)^T Q_{\text{obj}} \left(x_N^i - x_{\text{ref}}^i\right) + \mathbb{U}^T R_{\text{obj}} \mathbb{U}$$

with $Q_{\text{obj}} = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$, $R_{\text{obj}} = 0.001I$, x_k^i *is the state of vehicle* i *at the* k*-th time-step,* x_{ref}^i *is the reference state of vehicle* i*, and* $\mathbb{U}$ *is the concatenation of all the vehicle's control inputs.*

The starting and goal locations for each of the vehicles are

$$x_0^1 = \begin{bmatrix} -0.87 \\ 0.5 \\ 0 \\ 0 \end{bmatrix}, \quad x_0^2 = \begin{bmatrix} 0.87 \\ 0.5 \\ 0 \\ 0 \end{bmatrix}, \quad x_0^3 = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$x_{\text{ref}}^1 = \begin{bmatrix} 0.5 \\ 1.87 \\ 0 \\ 0 \end{bmatrix}, \quad x_{\text{ref}}^2 = \begin{bmatrix} -0.5 \\ 1.87 \\ 0 \\ 0 \end{bmatrix}, \quad x_{\text{ref}}^3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

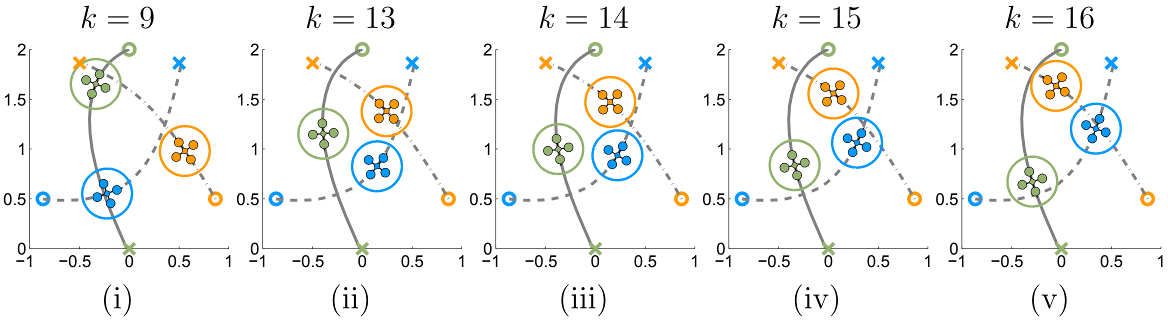
Due to the large number of feasible trajectories through the environment, the allowed feasible region was restricted to reduce the computation time. Consequently, only a locally optimal solution can be guaranteed which is the optimal solution within the restricted feasible region. The decomposed regions are defined by,

$$(a) \left\{ \begin{array}{l} \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} (x_k^1 - x_k^2) \leq -0.25 \\ \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} (x_k^1 - x_k^3) \leq -0.25 \quad k = 1, \ldots, k_1 \\ \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} (x_k^2 - x_k^3) \leq -0.25 \end{array} \right.$$

$$(b) \left\{ \begin{array}{l} \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} (x_k^1 - x_k^2) \leq -0.25 \\ \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} (x_k^3 - x_k^1) \leq -0.25 \quad k = k_1 + 1, \ldots, k_2 \\ \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} (x_k^3 - x_k^2) \leq -0.25 \end{array} \right.$$

$$(c) \left\{ \begin{array}{l} \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} (x_k^3 - x_k^1) \leq -0.25 \\ \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} (x_k^3 - x_k^2) \leq -0.25 \quad k = k_2 + 1, \ldots, N \\ \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} (x_k^2 - x_k^1) \leq -0.25 \end{array} \right.$$

where again k_1 and k_2 are the times when the vehicle exits and enters the different regions. In words, the constraints above are as follows: (a) requires vehicle 3 to be above vehicles 1, 2 and vehicle 1 to be to the left of vehicle 2, (b) requires vehicle 3 to



(i) (ii) (iii) (iv) (v)

Figure 4.5: A three-vehicle collision avoidance scenario solved using Boole's method with an allowed constraint violation of $\delta = 0.1$. In each plot, the vehicles are attempting to reach their goal location marked by the 'x'.

be to the left of vehicles 1, 2 and vehicle 2 to be above vehicle 1, (c) requires vehicle 3 to be below vehicles 1, 2 and vehicle 2 to be to the left of vehicle 1.

Only Boole's method is presented here, but the CVaR method and the ellipsoidal relaxation method produce qualitatively similar results. Figure 4.5 shows a series of images of the vehicle locations for the expected trajectories for a problem with an allowed constraint violation of $\delta = 0.1$. For each of the subplots in the figure, the system is in regions (a),(b),(b),(b),(c), respectively, of the decomposed environment.

4.6 Conclusions

This chapter presented three methods for solving the stochastic control problem with deterministic constraint parameters and they were evaluated on several examples of stochastic motion planning through known environments. It demonstrated that the inherit stochasticity in the problem can be directly handled without drastically increasing the computational complexity while rigorously providing the user's desired safety level. As already illustrated in Chapter 3, in the presented examples the ellipsoidal relaxation method is the most conservative, however, it has the lowest computational complexity. Boole's method is slightly more computationally complex, but generates the least conservative solution. The conservativeness is due to Boole's inequality, but the examples demonstrated that it was indeed small. This chapter also illustrated the applicability of the CVaR method to handle these types of problems.

Chapter 5

Feedback Design and Risk Allocation in Chance Constrained Control

In the previous chapter, the controller was assumed to be fixed and the chance constraints were satisfied by optimizing over the desired trajectory. By using a fixed controller, however, the system may be forced to take a longer, more costly route to satisfy the chance constraints. For Example 4, the chosen feedback controller resulted in a large vertical uncertainty in the system state which prevented the system from proceeding through the shorter bottom route. Since the distribution of the closed-loop system is dependent on the feedback controller parameters, the bottom route could have been enabled by modifying the controller.

This chapter incorporates feedback controller design into the stochastic control formulation, enabling the feedback controller to shape the uncertainty of the system to make feasible previously infeasible solutions. In prior work [28], an ellipsoidal relaxation technique was used to solve this problem. For this method, a static, uniform allocation of risk is assigned to each constraint transforming the problem into a deterministic program for optimizing the feedback parameters. However this results in a large amount of conservatism, degrading the performance of the overall system.

To reduce the conservatism, this chapter presents a novel approach for optimizing over both the feedback controller and the risk allocation of the joint chance constraints. Instead of using the ellipsoidal relaxation technique, the joint chance constraints are handled using either Boole's method or the CVaR method which results in a tighter approximation. Using the CVaR method, the resulting optimization program is convex and optimizes over both controller and risk allocation simultaneously. However, the computational complexity grows quickly with the problem size, rendering some problems intractable to solve using this method. Consequently, the focus of the chapter is on using Boole's method. Direct application of Boole's method yields an optimization problem that is not readily solvable. Therefore, a two-stage optimization algorithm is presented that alternates between optimizing the feedback controller and the risk allocation until convergence. This solution methodology is shown to reduce the conservatism in previous approaches and improve the performance of the overall system.

Example 4 presented in Section 4.5.2 is used to compare the proposed methods with the ellipsoidal relaxation method. Boole's method is shown to enable the previously infeasible path through the bottom route in this example, however, the ellipsoidal relaxation method still requires the system to take the suboptimal top route. For this example, the CVaR method proved too computationally complex to use.

5.1 System Description

The system evolves according to

$$\mathbb{X} = T^{xx_0}x_0 + T^{xu}\mathbb{U} + T^{xw}\mathbb{W} \tag{5.1}$$

$$\mathbb{Y} = C\mathbb{X} + T^{yv}\mathbb{V} \tag{5.2}$$

where in an abuse of notation $C = [\operatorname{diag}(C, \ldots, C)\, 0] \in \mathbb{R}^{pN \times n(N+1)}$ with diag(·) forming a block diagonal matrix from its arguments,

$$
\begin{aligned}
T^{xx_0} &= \begin{bmatrix} I \\ A_0^1 \\ A_0^2 \\ \vdots \\ A_0^N \end{bmatrix}, \quad
T^{xu} = \begin{bmatrix} 0 & & & & \\ A_1^1 B_0 & 0 & & & \\ A_1^2 B_0 & A_2^2 B_1 & 0 & & \\ \vdots & & & \ddots & \\ \vdots & & & & 0 \\ A_1^N B_0 & A_2^N B_1 & \ldots & \ldots & A_N^N B_{N-1} \end{bmatrix}, \\
T^{yv} &= \begin{bmatrix} V_0 \\ V_1 \\ \vdots \\ V_{N-1} \end{bmatrix}, \quad
T^{xw} = \begin{bmatrix} 0 & & & & \\ A_1^1 W_0 & 0 & & & \\ A_1^2 W_0 & A_2^2 W_1 & 0 & & \\ \vdots & & & \ddots & \\ \vdots & & & & 0 \\ A_1^N W_0 & A_2^N W_1 & \ldots & \ldots & A_N^N W_{N-1} \end{bmatrix},
\end{aligned}
\tag{5.3}
$$

where $A_\tau^k = A_{k-1} A_{k-2} \ldots A_\tau$ with $A_k^k = I$.

A feedback controller can be used to shape the uncertainty of the system state, facilitating the satisfaction of the chance constraints and improving the objective function cost. As has been considered in the past [28, 88], only the set of affine causal output feedback controllers will be considered here, i.e.,

$$
u_k = \bar{u}_k + \sum_{\tau=0}^{k} K_{k,\tau} y_\tau, \quad k = 0, \ldots, N-1. \tag{5.4}
$$

The control law can also be defined in terms of a reference output, y_τ^r, i.e. $u_k = \bar{u}_k + \sum_{\tau=0}^{k} K_{k,\tau} (y_\tau - y_\tau^r)$. The compact form of Eqn. (5.4) is given by,

$$
\mathbb{U} = \bar{\mathbb{U}} + K \mathbb{Y} \tag{5.5}
$$

where

$$K = \begin{bmatrix} K_{0,0} & 0 & \dots & 0 \\ K_{1,0} & K_{1,1} & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ K_{N-1,0} & K_{N-1,1} & \dots & K_{N-1,N-1} \end{bmatrix}, \tag{5.6}$$

and $\bar{\mathbb{U}} = \left[\bar{u}_0^{\mathrm{T}}, \dots, \bar{u}_{N-1}^{\mathrm{T}}\right]^{\mathrm{T}}$.

Now the characteristics of the closed-loop state and input can be determined. Substituting for $\mathbb{X}$ from Eqn. (5.2) yields,

$$\mathbb{Y} = C\left(T^{xx_0}x_0 + T^{xu}\mathbb{U} + T^{xw}\mathbb{W}\right) + T^{yv}\mathbb{V}. \tag{5.7}$$

Substituting for $\mathbb{U}$ from Eqn. (5.5) results in,

$$\mathbb{Y} = C\left(T^{xx_0}x_0 + T^{xu}\left(\bar{\mathbb{U}} + K\mathbb{Y}\right) + T^{xw}\mathbb{W}\right) + T^{yv}\mathbb{V}. \tag{5.8}$$

After simplifying and combining common terms,

$$\left(I - CT^{xu}K\right)\mathbb{Y} = C\left(T^{xx_0}x_0 + T^{xu}\bar{\mathbb{U}} + T^{xw}\mathbb{W}\right) + T^{yv}\mathbb{V}. \tag{5.9}$$

The term $CT^{xu}K$ is lower triangular, therefore $I - CT^{xu}K$ is invertible. Solving for $\mathbb{Y}$ yields,

$$\mathbb{Y} = \left(I - CT^{xu}K\right)^{-1}\left(C\left(T^{xx_0}x_0 + T^{xu}\bar{\mathbb{U}} + T^{xw}\mathbb{W}\right) + T^{yv}\mathbb{V}\right). \tag{5.10}$$

Substituting this into Eqn. (5.5) and simplifying yields,

$$\begin{aligned} \mathbb{U} = \; & K\left(I - CT^{xu}K\right)^{-1}CT^{xx_0}x_0 + \left(I + K\left(I - CT^{xu}K\right)^{-1}CT^{xu}\right)\bar{\mathbb{U}} + \\ & K\left(I - CT^{xu}K\right)^{-1}CT^{xw}\mathbb{W} + K\left(I - CT^{xu}K\right)^{-1}T^{yv}\mathbb{V}. \end{aligned} \tag{5.11}$$

Substituting this into Eqn. (5.1) results in,

$$\begin{aligned}\mathbb{X} = \;& \left(I + T^{xu}K\left(I - CT^{xu}K\right)^{-1}C\right)T^{xx_0}x_0 + T^{xu}\left(I + K\left(I - CT^{xu}K\right)^{-1}CT^{xu}\right)\bar{\mathbb{U}} + \\ & \left(I + T^{xu}K\left(I - CT^{xu}K\right)^{-1}C\right)T^{xw}\mathbb{W} + T^{xu}K\left(I - CT^{xu}K\right)^{-1}T^{yv}\mathbb{V}.\end{aligned} \tag{5.12}$$

Youla Parametrization

In the current problem formulation, the controller gains, K, are being optimized in order to shape the covariance to satisfy the constraints and lower the objective cost. In general, the program is not convex in the original design variables K and $\bar{\mathbb{U}}$. However, [28, 88] showed by a change of variables the problem can be cast as a convex optimization problem. This is accomplished by using the Youla parametrization as follows,

$$Q = K\left(I - CT^{xu}K\right)^{-1}. \tag{5.13}$$

The original gain matrix K can then be efficiently solved for by

$$K = \left(I + QCT^{xu}\right)^{-1}Q, \tag{5.14}$$

resulting in a block lower triangular matrix. Also, define

$$r = \left(I + QCT^{xu}\right)\bar{\mathbb{U}} \tag{5.15}$$

such that

$$\bar{\mathbb{U}} = \left(I + QCT^{xu}\right)^{-1}r = \left(I + KCT^{xu}\right)r. \tag{5.16}$$

The variables Q and r can now be used in place of K and $\bar{\mathbb{U}}$ to generate a convex optimization program. Using this change of variables, the control input and state simplify to

$$\mathbb{U} = QCT^{xx_0}x_0 + r + QCT^{xw}\mathbb{W} + QT^{yv}\mathbb{V}, \tag{5.17}$$

$$\mathbb{X} = \left(I + T^{xu}QC\right)T^{xx_0}x_0 + T^{xu}r + \left(I + T^{xu}QC\right)T^{xw}\mathbb{W} + T^{xu}QT^{yv}\mathbb{V}, \tag{5.18}$$

which are affine expressions in Q and r. Similarly, the uncertainty of the state and control input can be expressed in terms of Q and r,

$$\Sigma_{\mathbb{X}} = LT^{xx_0}\Sigma_0 {T^{xx_0}}^{\mathrm{T}} L^{\mathrm{T}} + LT^{xw}\Sigma_{\mathbb{W}} {T^{xw}}^{\mathrm{T}} L^{\mathrm{T}} + T^{xu}QT^{yv}\Sigma_{\mathbb{V}} {T^{yv}}^{\mathrm{T}} Q^{\mathrm{T}} {T^{xu}}^{\mathrm{T}}, \tag{5.19}$$

$$\Sigma_{\mathbb{U}} = QCT^{xx_0}\Sigma_0 {T^{xx_0}}^{\mathrm{T}} C^{\mathrm{T}} Q^{\mathrm{T}} + QCT^{xw}\Sigma_{\mathbb{W}} {T^{xw}}^{\mathrm{T}} C^{\mathrm{T}} Q^{\mathrm{T}} + QT^{yv}\Sigma_{\mathbb{V}} {T^{yv}}^{\mathrm{T}} Q^{\mathrm{T}}, \tag{5.20}$$

where $L = I + T^{xu}QC$.

Using the above formulations, the stochastic control problem with feedback design reduces to the optimization program (5.21).

$$\begin{array}{ll}
\underset{Q,r}{\text{minimize}} & \mathbf{E}\left[\phi(\mathbb{X},\mathbb{U})\right] \\
\text{subject to} & \\
& \mathbb{X} = \left(I + T^{xu}QC\right)T^{xx_0}x_0 + T^{xu}r + \left(I + T^{xu}QC\right)T^{xw}\mathbb{W} + T^{xu}QT^{yv}\mathbb{V} \\
& \mathbb{U} = QCT^{xx_0}x_0 + r + QCT^{xw}\mathbb{W} + QT^{yv}\mathbb{V} \\
& \mathbb{W} \sim \mathcal{N}(0,\Sigma_w) \\
& \mathbb{V} \sim \mathcal{N}(0,\Sigma_v) \\
& \bar{\mathbb{U}} \in F_{\mathbb{U}} \\
& \Sigma_{\mathbb{X}} = LT^{xx_0}\Sigma_0 {T^{xx_0}}^{\mathrm{T}} L^{\mathrm{T}} + LT^{xw}\Sigma_{\mathbb{W}} {T^{xw}}^{\mathrm{T}} L^{\mathrm{T}} + T^{xu}Q\Sigma_{\mathbb{V}} Q^{\mathrm{T}} {T^{xu}}^{\mathrm{T}} \\
& L = I + T^{xu}QC \\
& \mathrm{P}(H^{\mathrm{T}}\mathbb{X} - b > 0) \leq \delta \\
& Q \text{ lower triangular}
\end{array} \tag{5.21}$$

The primary complication in solving the optimization program (5.21) is in evaluating and satisfying the state chance constraints. While they may first appear to be easy to evaluate, the controller parameters can now affect the covariance matrix, adding further complexity into satisfying the chance constraints and solving the optimization program. Fortunately, there are many different ways to simplify the problem, presented in the next section.

5.2 Handling the Chance Constraints with Feedback Design

Previously when optimizing over the controller feedback gains ([28, 29]), an ellipsoidal relaxation technique was used to simplify the constraints, typically leading to a very conservative approximation of the probability of failure which increases the overall objective cost. Another approach to simplify the chance constraints is to use Boole's inequality which provides a tighter approximation of the probability of failure at the expense of a more complicated optimization program. The application of the CVaR method to this problem formulation is straightforward and no modification of the algorithm is needed. The subsections below describe how to apply the original formulation of Boole's and the ellipsoidal relaxation approaches presented in Chapter 3 to the feedback design problem.

Ellipsoidal Relaxation

In the ellipsoidal relaxation method, the original constraint $\mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} \leq b\right) \geq 1-\delta$ is replaced by requiring $\bar{\mathbb{X}} + \mathcal{E}_r \subset F_z$ which is equivalent to requiring

$$h_i^{\mathrm{T}}\bar{\mathbb{X}} + r\sqrt{h_i^{\mathrm{T}}\Sigma_{\mathbb{X}}h_i} \leq b_i, \quad i = 1,\ldots,q, \tag{5.22}$$

where h_i is the i-th column of H. These constraints form a set of second order cone constraints.

Boole's Inequality

Chapter 3 showed that the joint chance constraint $\mathrm{P}(H^{\mathrm{T}}\mathbb{X} \leq b)$ can be converted into univariate integrals by using Boole's inequality to conservatively bound the probability of violation. If the constraints

$$\begin{gathered} 1 - \Phi\left(\frac{b_i - h_i^{\mathrm{T}}\bar{\mathbb{X}}}{\sqrt{h_i^{\mathrm{T}}\Sigma_{\mathbb{X}}h_i}}\right) \leq \delta_i \\ \sum \delta_i \leq \delta \end{gathered} \tag{5.23}$$

are satisfied then the original joint chance constraint will also be satisfied.

In the current problem formulation, both δ_i and $\Sigma_{\mathbb{X}}$ are variables and there is no tractable way of simultaneously optimizing over both. However, if either variable is fixed then the optimization problem can be solved efficiently. This property is exploited in the proposed solution methodology in Section 5.3 below.

For a fixed covariance, $\Sigma_{\mathbb{X}}$, and $\delta \leq 0.5$ the constraints in Eqn. (5.23) are convex since the function $\Phi(x)$ is concave in the range $x \in [0, \infty)$ [31, 70]. If the risk allocation, δ_i, is fixed such that $\sum \delta_i \leq \delta$ then the constraints in Eqn. (5.23) can be converted into equivalent second order cone constraints as follows. The constraints require

$$1 - \Phi\left(\frac{b_i - h_i^{\mathrm{T}}\bar{\mathbb{X}}}{\sqrt{h_i^{\mathrm{T}}\Sigma_{\mathbb{X}} h_i}}\right) \leq \delta_i, \tag{5.24}$$

which can be simplified to

$$h_i^{\mathrm{T}}\bar{\mathbb{X}} + \Phi^{-1}\left(1 - \delta_i\right)\sqrt{h_i^{\mathrm{T}}\Sigma_{\mathbb{X}} h_i} \leq b_i \tag{5.25}$$

where Φ^{-1} is the inverse of the Gaussian cumulative distribution function. These constraints are similar to the constraint for the ellipsoidal relaxation method in Eqn. (5.22), however, the back-off parameter for Boole's method ($\Phi^{-1}(1 - \delta_i)$) is less conservative. Finally, the standard deviation of the constraint uncertainty can be converted into a second order cone constraint,

$$\sqrt{h_i^{\mathrm{T}}\Sigma_{\mathbb{X}} h_i} = \left\| \left[\left(I + T^{xu}QC\right) T^{xx_0}\Sigma_0^{\frac{1}{2}}, (I + T^{xu}QC)T^{xw}\Sigma_{\mathbb{W}}^{\frac{1}{2}}, T^{xu}QT^{yv}\Sigma_{\mathbb{V}}^{\frac{1}{2}} \right] h_i \right\|. \tag{5.26}$$

Boole's or Ellipsoidal Relaxation Method Optimization Program

Using either method for bounding the chance constraints, the program (5.21) can be simplified to (5.27).

$$
\begin{array}{ll}
\underset{Q,r}{\text{minimize}} & \mathbf{E}\left(f(\mathbb{X},\mathbb{U})\right) \\
\text{subject to} & \\
& \mathbb{X} = \left(I + T^{xu}QC\right)T^{xx_0}x_0 + T^{xu}r + \left(I + T^{xu}QC\right)T^{xw}\mathbb{W} + T^{xu}QT^{yv}\mathbb{V} \\
& \mathbb{U} = QCT^{xx_0}x_0 + r + QCT^{xw}\mathbb{W} + QT^{yv}\mathbb{V} \\
& \mathbb{W} \sim \mathcal{N}(0,\Sigma_w) \\
& \mathbb{V} \sim \mathcal{N}(0,\Sigma_v) \\
& \bar{\mathbb{U}} \in F_{\mathbb{U}} \\
& L = I + T^{xu}QC \\
& \nu_i = \left\| \left[LT^{xx_0}\Sigma_0^{\frac{1}{2}}, LT^{xw}\Sigma_{\mathbb{W}}^{\frac{1}{2}}, T^{xu}QT^{yv}\Sigma_{\mathbb{V}}^{\frac{1}{2}} \right] h_i \right\| \\
& h_i^{\mathrm{T}}\bar{\mathbb{X}} + \beta_i\nu_i \le b_i, \quad i = 1,\ldots,q \\
& Q \text{ lower triangular}
\end{array}
\tag{5.27}
$$

In the optimization program (5.27), $\beta_i = r$ for the ellipsoidal relaxation technique, or $\beta_i = \Phi^{-1}(1-\delta_i)$ for Boole's method. For the ellipsoidal relaxation method the parameters β_i are fixed (given in Eqn. (3.9)), and the resulting optimization program (5.27) is a convex Second Order Cone Program (SOCP). The fixed parameters β_i, however, will be shown to lead to conservative solutions. For the case of Boole's method, both β_i and K are optimized, reducing the conservativeness of the resulting solutions. Optimizing over both the controller parameters and the risk allocation simultaneously is not a straightforward procedure, since β_i and ν_i are both involved in a multiplicative constraint. However, given a fixed controller K or fixed risk allocation δ_i, the program (5.27) is a convex optimization problem which can be efficiently solved. Using these facts, the next section presents a two-stage optimization method.

5.3 Boole's Method Solution Methodology: Two-Stage Method

Given the complexity in simultaneously optimizing the risk allocation as well as the controller parameters for Boole's method, an iterative two-stage optimization scheme is utilized for solving program (5.27). The upper stage allocates the risk associated to each individual constraint while the lower stage solves a SOCP for the optimal controller parameters given the current risk allocation. The primary difficulty in solving this problem is devising the iterative risk allocation scheme for the upper stage.

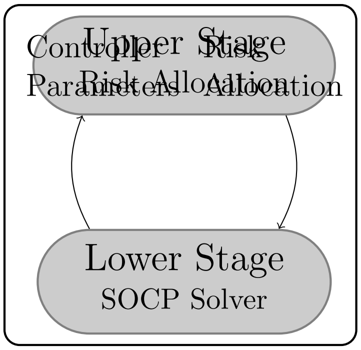


Figure 5.1: Two-stage optimization algorithm.

5.3.1 Upper Stage

There are several heuristic methods that can be devised for allocating the risk in the upper stage. This subsection describes two such methods in further detail: the bisection method and the optimal risk given a fixed controller method.

Bisection Method

The simplest heuristic to use is the bisection method presented in Algorithm 1, which adjusts the uniform allocation of the risk until the actual risk is within a specified threshold of the allowed value. The initialization of the method is described in Algorithm 2. First, two uniform risk allocations are computed (in lines 3 and 6 of

Algorithm 2) above and below the desired probability of constraint violation. A uniform risk allocation at the midpoint of these two allocations is then used to solve for the controller parameters in the lower stage. The solution from the lower stage is then used in Algorithm 1 to adjust the risk. If the true probability of constraint violation (calculated in line 1 of Algorithm 1) is less than the allowed risk, then it replaces the current lower bound, otherwise it replaces the upper bound. This is continued until the actual risk is within a threshold of the allowed risk.

Algorithm 1 Bisection Method for Risk Allocation

```
 1: $\delta_{i_{\text{true}}} = \sum_i 1 - \Phi\left(\frac{b_i - h_i^{\mathrm{T}}\bar{\mathbb{X}}}{\sqrt{h_i^{\mathrm{T}}\Sigma_{\mathbb{X}}h_i}}\right)$
 2: if $|\delta_{i_{\text{true}}} - \delta| \leq \epsilon$ then
 3:     Solution found.
 4: end if
 5: if $\delta_{i_{\text{true}}} \geq \delta$ then
 6:     $\overline{\delta} = \delta_i$
 7: else
 8:     $\underline{\delta} = \delta_i$
 9: end if
10: $\delta_i = 0.5(\underline{\delta} + \overline{\delta})$
```

Algorithm 2 Initialization for the Bisection Method

```
 1: $\underline{\delta} = \delta/q$
 2: Perform optimization of (5.27) with $\beta_i = \Phi^{-1}(1 - \underline{\delta})$
 3: $\underline{\delta}_{\text{true}} = \sum_i 1 - \Phi\left(\frac{b_i - h_i^{\mathrm{T}}\bar{\mathbb{X}}}{\sqrt{h_i^{\mathrm{T}}\Sigma_{\mathbb{X}}h_i}}\right)$
 4: $\overline{\delta} = 0.5$
 5: Perform optimization of (5.27) with $\beta_i = \Phi^{-1}(1 - \overline{\delta})$
 6: $\overline{\delta}_{\text{true}} = \sum_i 1 - \Phi\left(\frac{b_i - h_i^{\mathrm{T}}\bar{\mathbb{X}}}{\sqrt{h_i^{\mathrm{T}}\Sigma_{\mathbb{X}}h_i}}\right)$
 7: if $\overline{\delta}_{\text{true}} < \delta$ then
 8:     Solution found, no need to iterate.
 9: end if
10: $\delta_i = 0.5(\underline{\delta} + \overline{\delta})$
```

Optimal Risk Given Fixed Controller

As was stated before, if the controller parameters, K or Q, are fixed (and hence the covariance $\Sigma_{\mathbb{X}}$ is fixed) then the problem simplifies to a convex optimization program. Consequently, for the case of a fixed controller, the optimal risk allocation, $\delta_i \; \forall i$, for the particular controller parameters can be efficiently computed. However, there is no guarantee that this risk allocation is the optimal risk allocation for the original problem.

5.3.2 Lower Stage

Once the risk allocation has been fixed, the program (5.27) simplifies to a standard second order cone program which can be solved efficiently by many standard solvers.

5.4 Application: Motion Planning

The feedback design and risk allocation methods presented in this chapter are applicable to many different fields and is evaluated here on a variant of the stochastic motion planning problem in Example 4. As shown in Figure 4.4, by using a fixed LQR controller and Kalman filter estimator, the vehicle was forced to use the top route because the bottom route was infeasible due to violating the joint chance constraints. By instead optimizing over the feedback controller parameters, the uncertainty of the system can be shaped to enable the previous infeasible solution.

Example 6. *The following example is similar to Example 4 except for a* $Q_{obj} = \begin{bmatrix} 10I & 0 \\ 0 & 0 \end{bmatrix}$ *and* $\delta = 0.15$*. Here the* δ *was chosen to highlight the differences between Boole's and the ellipsoidal relaxation methods.*

The results are shown in Figure 5.2. The solution with a fixed LQR trajectory tracking controller and Kalman filter from Chapter 4 is shown in Figure 5.2(a) with an objective function cost of 0.184. Here, the system had to take the suboptimal route through the top; the bottom path was infeasible due to the large vertical uncertainty.

The solution using feedback design with the ellipsoidal relaxation technique is shown in Figure 5.2(b). Here, the system also has to take the suboptimal route through the top of the environment because the relaxation method for the joint chance constraints is too conservative. The objective function cost was 0.186 with a probability of failure of 0.004 which is significantly less than the allowed risk.

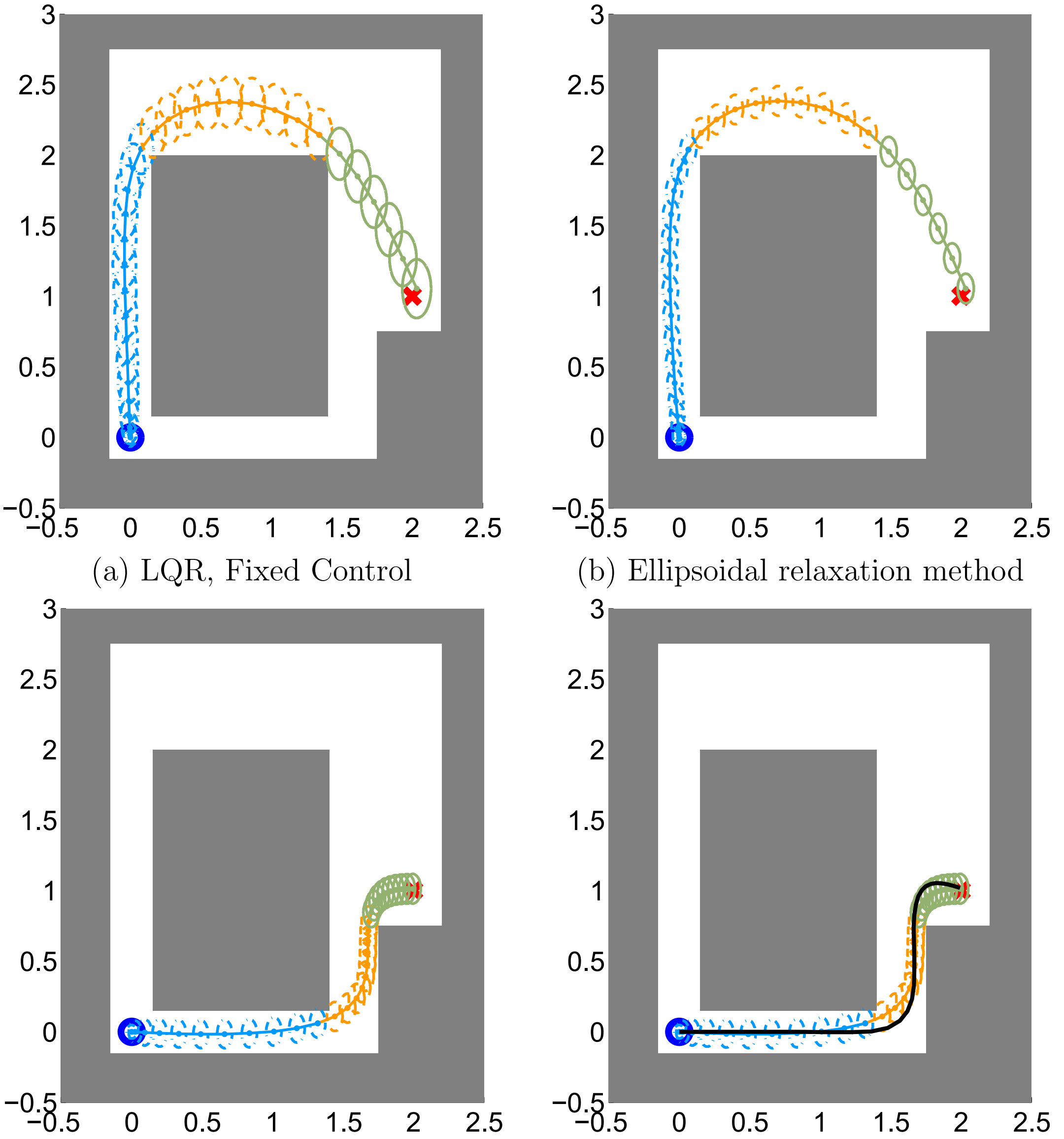
The solution using the new two-stage algorithm presented in this chapter for optimizing the output feedback controller parameters is shown in Figure 5.2(c-d). Figure 5.2(c) shows the solution using the binary search upper stage algorithm. The algorithm took 16 iterations for an $\epsilon = 1 \times 10^{-5}$ and the objective function's value was 0.120. In contrast, the solution using the optimal risk given fixed controller upper stage algorithm, shown in Figure 5.2(d), only took 1 iteration. For this solution, the upper stage was initialized with a uniform allocation of the risk $\delta_i = \delta/q$. The trajectory for this allocation is shown as the black line. The final solution is shown as the blue line with the 90% confidence ellipsoids around it. The initial objective cost is 0.146 and the final objective cost is 0.116 which is a 26% relative improvement.

The CVaR method was also used to solve this problem, however, the optimization program was tractable for only a very small number of particles and the solution always violated the joint chance constraints during cross validation.

A comparison of the computation times for the various methods is shown in Table 5.1. The computations were done using `CVX` [89] on a 2.7 GHz Intel Core i7. The ellipsoidal relaxation algorithm is the fastest algorithm. The optimal risk two-stage optimization method is 46% slower but results in a 60% better objective function value. The binary search method is impractical to solve this particular problem, as its computation time is over 10 times slower than the optimal risk method.

Table 5.1: A comparison of the computation times and objective function value for the different solution methods.

	Comp. Time (secs.)	Objective
Ellipsoidal relaxation method	56	0.186
Optimal risk method	82	0.116
Binary search method	900	0.120
CVaR method	–	–



(a) LQR, Fixed Control

(b) Ellipsoidal relaxation method

(c) Bisection Method

(d) Optimal Risk Upper Stage Method

Figure 5.2: The solution and the confidence ellipsoids for a nonconvex environment for Example 6. (a) Solution for an assumed LQR trajectory tracking controller as used in Example 4. (b) The solution for the ellipsoidal relaxation method. (c) The solution from the two stage algorithm using the bisection method for the upper stage. (d) The solution for the optimal risk given fixed controller upper stage. The black line is the solution from the SOCP solver assuming a uniform allocation of risk. The blue line, with the confidence ellipsoids around it, is the solution using the controller parameters from the first SOCP stage and optimizing the risk for each constraint.

5.5 Conclusions

This chapter studied the problem of simultaneous feedback and risk allocation in chance constrained control. Previously in Chapter 4, a chance constrained stochastic control problem was investigated in which a fixed controller and estimator were employed, and only the risk of violating each constraint was optimized. However, this chapter showed that a better solution can be found by also optimizing over the feedback parameters.

Currently, the method is only suitable for offline computations, however, real-time operation could be employed by attempting to exploit structure in the optimization program as well as by reducing the size of the feedback controller. One way to reduce the size of the controller is to only use a subset of the previous measurements to compute the control input, which does not affect the convexity of the problem. Even without real-time operation, the presented method has the potential to have a large impact on the performance of many systems.

Chapter 6

Stochastic Control for Jump Markov Linear Systems

The previous chapters demonstrated how to handle the joint chance constraints using both analytical and sampling methods. The analytical methods assume the distribution of the system can be represented well by a Gaussian distribution. While this assumption is valid for many of the motivating applications as presented in the previous chapters, there are cases for which this assumption does not hold. One such example in which the probability distribution of the system is non-Gaussian is in jump Markov linear systems, which consist of multiple discrete modes, each with different continuous dynamics [90]. For these systems, the stochastic nature of the discrete state results in a multimodal distribution of the future system state. In these cases, the sampling methods are advantageous over the analytical methods due to their ability to handle arbitrary distributions.

Jump Markov linear systems have many applications including modeling systems with component failures. Another example, from the motivating applications in this thesis, is in autonomous control of vehicles. To safely operate on multiple lane highways, an autonomous vehicle must maintain a probability distribution of other cars' lateral position in order to make safe passing decisions. This distribution is dependent on several factors including the driver intent. One way of modeling this is to use a jump Markov linear system with different probability distributions based on the state of the blinker of other cars. A blinker on would indicate a high probability that a vehicle will change into the indicated lane, and a low probability of changing into the opposite lane. This scenario leads to an arbitrary, multimodal distribution which is difficult to capture analytically and thus requires the use of sampling methods as described in this chapter.

The feedback control problem for jump Markov linear systems has been extensively studied [90]. This previous work was primarily concerned with solving the feedback control problem to minimize the expected value of a cost function given the value of the state and mode at the current time-step. In contrast, this chapter is concerned with the predictive stochastic control problem which takes into account the future distribution of the system in order to control it to satisfy the system's constraints. Using this formulation, Blackmore et al. [91] proposed a stochastic control algorithm via sampling which used the Mixed Integer Programming (MIP) method to handle the chance constraints, however, the resulting optimization problem is nonconvex. This solution technique is NP-hard in the number of binary variables used in the problem formulation [92, 93], and consequently the computational requirements grow exponentially as the number of samples needed to model the problem increases.

This chapter proposes using the CVaR method rather than MIP to handle the chance constraints in the stochastic control problem for jump Markov linear systems. The resulting optimization program is convex, leading to faster solutions than MIP. To highlight the differences between the MIP and CVaR methods, they are compared in a vehicle planning problem involving brake failures. In this example, while both methods yield nearly identical planning results, the CVaR method is shown to reduce the computation time by a factor of 10 over MIP.

6.1 System Description

In this chapter, the system dynamics are described by a jump Markov linear system formulated as a discrete-time stochastic hybrid system. The system state at time-step k contains both a continuous state x_k and a discrete mode σ_k. The continuous state and output of the stochastic switched system evolve according to

$$x_{k+1} = A_{\sigma_k} x_k + B_{\sigma_k} u_k + W_{\sigma_k} w_k, \tag{6.1}$$

$$y_k = C_{\sigma_k} y_k + V_{\sigma_k} v_k, \tag{6.2}$$

where w_k and v_k are the process and output noises with known probability distributions $p(w_k)$ and $p(v_k)$. The discrete mode $\sigma_k \in \{1, \ldots, M\}$ is a Markov chain that evolves according to

$$\mathrm{P}(\sigma_{k+1} = j | \sigma_k = i) = T_{ij}, \tag{6.3}$$

where $T \in \mathbb{R}^{M \times M}$ is the transition matrix. The initial distribution of the continuous-discrete state is assumed to be known and given by $p(x_0, \sigma_0)$. The discrete mode is assumed to be independent of the continuous state and continuous input.

Given the independence of the discrete and continuous states, the probability distribution of the hybrid continuous-discrete state is

$$\mathrm{P}\left(\mathbb{X}, \boldsymbol{\sigma} | \mathbb{U}, \mathbb{Y}\right) = \mathrm{P}\left(\mathbb{X} | \boldsymbol{\sigma}, \mathbb{U}, \mathbb{Y}\right) \mathrm{P}\left(\boldsymbol{\sigma}\right), \tag{6.4}$$

where $\boldsymbol{\sigma} = [\sigma_0, \ldots, \sigma_N]^{\mathrm{T}}$ and N is the finite time horizon. In general, even if all the continuous distributions, $\mathrm{P}\left(\mathbb{X} | \boldsymbol{\sigma}, \mathbb{U}, \mathbb{Y}\right)$, are Gaussian, the total distribution is a multimodal, non-Gaussian distribution due to mode switching. These types of systems illustrate the benefit of the sampling methods introduced in Chapter 3 over the analytical methods. While the sampling methods are able to handle arbitrary distributions, the analytical methods use a Gaussian distribution which poorly models the true distribution, leading to violation of the joint chance constraints.

The following section reviews the aspects of random variable sampling that are most relevant to applying the sampling methods for stochastic control of jump Markov linear systems.

6.2 Sampling from Random Variables

The typical way of dealing with non-parametric probability distributions is to approximate them with samples drawn from the distribution, instead of using an analytical function [94]. This technique has lead to many tractable algorithms in estimation and control. The properties of standard and importance sampling strategies are reviewed below.

Standard Sampling Strategy

As presented in Section 3.2, a multivariate random variable z with known probability distribution $p(z)$ can be approximated by drawing N_s independent, identically distributed random samples:

$$\left\{z^{(1)}, \ldots, z^{(N_s)}\right\}. \tag{6.5}$$

The samples can be used to efficiently evaluate expressions involving the random variable. For stochastic control, the two relevant expressions are the expectation and the probability of an event. The expectation can be efficiently approximated using the sample mean.

$$\mathrm{E}\left[f\left(z\right)\right] = \int_z f\left(z\right) p(z) dz \approx \frac{1}{N_s} \sum_{i=1}^{N_s} f\left(z^{(i)}\right) \tag{6.6}$$

As the number of samples tends to infinity, the sample mean will converge to the true value of the expectation.

In chance constrained control, the samples are also used to approximate the probability of an event occurring in the joint chance constraints, i.e.,

$$\mathrm{P}\left(\psi(z) > 0\right) = \int_z \mathbf{1}(\psi(z) > 0) p(z) dz \approx \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{1}(\psi(z^{(i)}) > 0). \tag{6.7}$$

Again, as the number of samples tends to infinity, this approximation approaches the true probability of an event occurring. Consequently, if a large number of samples is used, then there is a high confidence that the approximation of the event probability is close to the true probability. However, representing low probability regions of the underlying distribution requires a large number of samples, which for some applications is intractable. Indeed, this is shown to be the case for the standard MIP approach (formulated in Section 3.2.1) applied to stochastic control of jump Markov linear systems [91].

Importance Sampling Strategy

To deal with this case, importance sampling can be used in which samples are drawn from an alternative proposal distribution [94]. This method has the benefit of being able to choose where to draw samples, making it easier to represent all areas of the probability distribution with a small number of samples.

In importance sampling, N_s independent, identically distributed random samples $\left\{z^{(1)}, \dots, z^{(N_s)}\right\}$ are generated from the proposal distribution, $q(z)$, which is defined such that $p(z) > 0$ implies $q(z) > 0$. For each sample, an importance weight w_i is calculated via $w_i = p(z^{(i)})/q(z^{(i)})$. The expectation of a function is then calculated by

$$\mathrm{E}\left[f\left(z\right)\right] \approx \frac{1}{N_s} \sum_{i=1}^{N_s} w_i f\left(z^{(i)}\right). \tag{6.8}$$

In contrast to the standard sampling method, the samples have a non-uniform weighting. Similarly, the probability of the event occurring in the joint chance constraints is calculated by the weighted sum

$$\mathrm{P}\left(\psi(z) > 0\right) \approx \frac{1}{N_s} \sum_{i=1}^{N_s} w_i \mathbf{1}(\psi(z^{(i)}) > 0). \tag{6.9}$$

For the importance sampling method, both the expectation and the probability of an event occurring converge to the true value as the number of samples tends to infinity.

The next section describes the application of these sampling methods to stochastic control of jump Markov systems.

6.3 Stochastic Control using Sampling

The procedure for solving the chance constrained control problem using sampling is described in Algorithm 3 [91]. In Step 1, either the standard sampling or the importance sampling technique can be used to generate the samples. However, typically the importance sampling strategy is preferred as it can use fewer samples to accurately represent the distribution. The primary difference between the MIP and CVaR methods is in the evaluation of the joint chance constraints in Step 3. The details of these two methods is described below.

Algorithm 3 Stochastic Control via Sampling Algorithm

```
1: Generate $N_s$ samples from the distribution of the initial state, process and measurement noises, and discrete mode sequence.
2: Express the distribution of the future system states using the generated samples. The state trajectory is a function of the initial state, process and measurement noises, discrete mode sequence for that particular sample, as well as the control inputs, $\mathbb{U}$, that are applied to all samples.
3: Use the samples to evaluate and satisfy the joint chance constraints.
4: Approximate the objective function using the samples.
5: Solve the deterministic constrained optimization program for the control inputs $\mathbb{U}$.
```

Mixed Integer Programming

The MIP method approximates the joint chance constraints using importance sampling as follows,

$$\mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} - b > 0\right) \approx \frac{1}{N_s}\sum_{i=1}^{N_s} w_i \mathbf{1}\left(H^{\mathrm{T}}\mathbb{X}^{(i)} - b > 0\right) \leq \delta, \tag{6.10}$$

where $\mathbf{1}(\cdot)$ is the indicator function. Unfortunately, these constraints lead to a nonconvex, mixed integer program because the only way to represent the indicator function is by using binary variables.

The computational complexity of a MIP is exponential in the number of binary variables needed to express the optimization program. Since one binary is needed for each sample, importance sampling is required for this approach in order to make the chance constraint evaluation tractable.

CVaR Method

The CVaR method bounds the joint chance constraints by enforcing

$$\mathrm{E}\left[\left(\max(H^{\mathrm{T}}\mathbb{X} - b + \alpha)\right)_{+}\right] \leq \alpha\delta, \tag{6.11}$$

where the expectation is evaluated using importance sampling as follows

$$\frac{1}{N_s}\sum_{i=1}^{N_s} w_i \max\{\max(H^{\mathrm{T}}\mathbb{X}^{(i)} - b + \alpha), 0\}. \tag{6.12}$$

As was shown in Section 3.2.3, the constraint in Eqn. (6.11) is convex, and therefore the CVaR method has a drastically smaller computational complexity over the MIP method. This will be illustrated in example in the following section.

6.4 Vehicle Brake Failure Example

The example presented in [91] is used here to compare the MIP technique with the proposed CVaR method. In this example, a vehicle traveling in one dimension is attempting to park without colliding with a nearby wall, as shown in Figure 6.1. The car has two discrete modes: (1) the brakes are functional and can apply full control braking authority, and (2) the brakes have failed and have no control braking authority.

Example 7. *Consider a vehicle with dynamics given by*

$$x_{k+1} = \begin{bmatrix} 1 & 0.940 \\ 0 & 0.883 \end{bmatrix} x_k + B_{\sigma_k}(u_k + w_k), \tag{6.13}$$

where w_k is the process noise with the following distribution $w_k \sim \mathcal{N}(0, 1\times 10^{-5} I)$,

Figure 6.1: Illustration of the vehicle planning problem with break failure. The vehicle's objective is to park at the gray patch without colliding with the brick wall on the right.

and B_{σ_k} *can take on two different forms depending on the status of the brakes:*

$$B_{\sigma_k} = \begin{cases} \begin{bmatrix} 0.48 & -0.48 \\ 0.94 & -0.94 \end{bmatrix}, & \sigma_k = \textit{brakes ok} \\ \begin{bmatrix} 0.48 & 0 \\ 0.94 & 0 \end{bmatrix}, & \sigma_k = \textit{brakes faulty.} \end{cases} \tag{6.14}$$

The control input is composed of two quantities u_{power} *and* u_{brake}*:* $u_k = \begin{bmatrix} u_{\text{power}} \\ u_{\text{brake}} \end{bmatrix}$. *When the system is in mode 2, the brakes are not functional and therefore* u_{brake} *cannot affect the system dynamics. Both terms of the control input are constrained to be greater than zero:* $u_k \geq 0$. *Also, since the brakes can only decelerate the vehicle, the velocity is constrained to be greater than or equal to zero:* $\begin{bmatrix} 0 & 1 \end{bmatrix} x_k \geq 0$ *for all* $k = 1, \ldots, N$.

The discrete mode is a Markov chain with the following transition matrix

$$T = \begin{bmatrix} 0.999 & 0.001 \\ 0.0 & 1.0 \end{bmatrix}. \tag{6.15}$$

At every time-step the brakes have a 0.001 *probability of failing, and once the brakes have failed they cannot return to being functional.*

The system is deemed to have failed if the vehicle collides with the wall at any *time-step. This leads to the following joint chance constraint*

$$\mathrm{P}\left(\begin{bmatrix} 0 & 1 \end{bmatrix} x_k > 10, \quad k = 1, \ldots, N\right) \leq \delta. \tag{6.16}$$

The allowed probability of constraint violation is $\delta = 0.01$ *and the time horizon is* $N = 20$*. The objective function for this problem is*

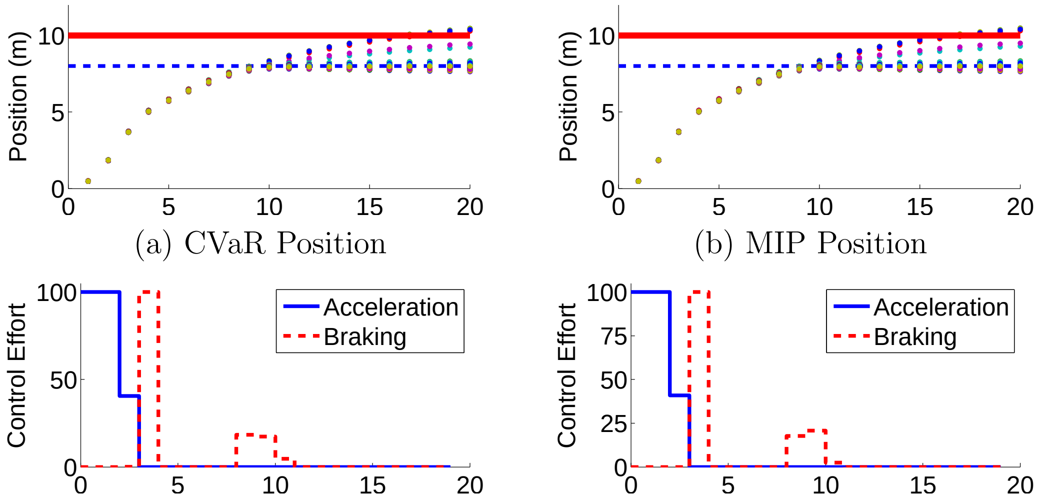
$$\phi(\mathbb{X}, \mathbb{U}) = \sum_{k=t_{\text{parked}}}^{N} \left(x_k - \begin{bmatrix} 8 \\ 0 \end{bmatrix}\right)^T \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \left(x_k - \begin{bmatrix} 8 \\ 0 \end{bmatrix}\right) + 0.1\mathbb{U}^T\mathbb{U} \tag{6.17}$$

where $t_{\text{parked}} = 10$ *is the time-step that the vehicle should be at the parking space.*

The MIP and CVaR methods are used to solve the resulting optimization program using $N_s = 1000$ samples to represent the uncertainty of the system. Figure 6.2 shows the resulting solution and control inputs for the two methods. Both solutions apply the maximum acceleration at the beginning and then apply the maximum braking force for one time-step. The only difference between the two solutions is a slight variation of the braking input applied between time-steps 7 to 10. After time-step 10, no control input is applied to the system.

As depicted in Figures 6.2(a-b), this control scheme results in three distinct sets of trajectories. One set of trajectories results in colliding with the wall due to a brake failure before time-step 4. The second set of trajectories overshoots the parking space, but avoids colliding with the wall. The final set of trajectories safely reaches the desired parking space at 8 meters.

A comparison of the computation time and the simulated true violation probability for the MIP and CVaR methods is shown in Table 6.1. The computation time is the average of 100 different iterations of each algorithm. The true violation probability was calculated through Monte Carlo simulations for the solution shown in Figure 6.2. The MIP method is more than 10 times slower computationally than the CVaR method. This is to be expected since the MIP method uses binary variables (the computational complexity of a MIP is exponential in the number of binary variables) while the CVaR method is a convex optimization program. The true violation



(a) CVaR Position

(b) MIP Position

(c) CVaR Control Effort

(d) MIP Control Effort

Figure 6.2: Results for the car planning failure example with the CVaR and MIP methods. (a)-(b) The samples of the position state for the CVaR and MIP methods. The blue dotted line is the location of the parking space. (c)-(d) The percentage of maximum control effort for the CVaR and MIP methods.

probability for both methods is almost identical.

Table 6.1: Timing results for the jump Markov linear system example

Method	time (secs.)	true δ
CVaR	21.6	0.0080
MIP	237.4	0.0078

6.5 Conclusions

This chapter studied the problem of stochastic control of jump Markov linear systems. Due to the inherent hybrid system description, the probability distribution of the system is poorly represented as a Gaussian distribution. Consequently, only the sampling methods are applicable for handling the joint chance constraints. In prior work, the MIP technique was used to solve this problem, however, it was shown that the CVaR method is also applicable to this problem. The optimization problem for the CVaR method is convex, resulting in a significant reduction in the computational complexity over the MIP technique. The computational benefit of CVaR over MIP was demonstrated through a vehicle planning example with brake failure.

In the following chapters, the framework is extended to handle uncertainty in the constraint parameters. This adds further complexity to the problem solution because the probability distribution is now the sum of products of random variables. In Chapter 7, three methods are explored for handling the joint chance constraints: Gaussian approximation method, CVaR method, and a new hybrid method. In Chapter 8, a dual control method is proposed to incorporate external sensing of the constraint parameters. This method trades off between completing its objective and exploring to gain more information to better complete its objective. Finally, in Chapter 9 the dual control stochastic control algorithm is experimentally demonstrated on a quadrotor unmanned aerial vehicle for a stochastic motion planning problem.

Chapter 7

Stochastic Control with Uncertain Constraint Parameters

The previous chapters demonstrated the success of incorporating the uncertainty of the system into the stochastic control problem formulation and solving the resulting optimal control problem. In all of the examples presented thus far, the constraint parameters were assumed to be deterministic, however, in general the constraints may themselves also have some uncertainty. An example of when this might occur is in robotic navigation. After a natural disaster, a robot used in a search and rescue operation might use original blueprints of a building as a general navigation guide, while the disaster could have caused significant changes to the building structure. Another example is navigating through moving obstacles that can be affected by unknown disturbances.

This chapter and the next increase the applicability of the proposed framework by extending it to handle uncertain constraint parameters. This extension presents several challenges. If all of the affine constraint parameters are allowed to be uncertain, then the general probability distribution will be summations of products of random variables. Typically, there is no analytic, closed-form function for the resulting probability distribution of the constraint, making it difficult to apply some of the solution methods presented in the earlier chapters. In addition, while future measurements of the system state can easily be accounted for in the current formulation, if the system can also take measurements of the uncertain constraint parameters the resulting problem formulation is a dual control problem which in general is intractable.

This chapter focuses on solving the stochastic control problem with uncertain constraints without the ability to directly sense the constraint parameters. Fortunately, for low-dimensional systems the constraints are shown to be approximated well by a univariate Gaussian distribution and can be evaluated efficiently. Boole's method can then use this approximate distribution to solve the stochastic control problem. Unfortunately, the resulting optimization program is nonconvex and therefore only a locally optimal solution can be guaranteed. In addition, the resulting solution is more conservative than for deterministic constraint parameters due to the increased correlation between time-steps which is ignored in Boole's inequality. However, the solutions are still reasonable.

In addition, the CVaR method will be shown to be applicable to solving these problems, handling the increased complexity of the uncertain parameters through sampling. However, the increased complexity of the resulting optimization program yields large computation times, restricting this method from being used in real-time applications.

A novel hybrid approach is also developed in this chapter, using a combination of sampling and analytic functions to represent the probability distributions of the system and constraint parameters. This method proves to be superior over the individual Boole's and CVaR method approaches. The hybrid approach results in a convex optimization program under certain conditions, guaranteeing the optimal solution. It also decreases the conservativeness of the solution over the approximation method. As compared to the CVaR method, typically fewer samples are needed which reduces the computational complexity and enables real-time operation.

Chapter 8 will investigate a tractable way of solving the dual control problem to enable the incorporation of new sensing of the constraint parameters into the problem solution.

7.1 System Description

In this chapter, the initial state, process noise and measurement noise are assumed to be independent Gaussian random variables distributed according to

$$x_0 \sim \mathcal{N}(\bar{x}_0, \Sigma_0), \quad w_k \sim \mathcal{N}(0, \Sigma_w) \quad \forall k, \quad v_k \sim \mathcal{N}(0, \Sigma_v) \quad \forall k. \tag{7.1}$$

As in Chapter 4, a Kalman filter is used to estimate the system state and the system is controlled via a linear quadratic trajectory tracking controller. The properties of the closed-loop system state and control input were derived in Section 4.1 and shown to be described by a Gaussian distribution with known mean and covariance. This will be exploited below in the evaluation of the chance constraints.

In the previous chapters, the constraint parameters H and b were assumed to be deterministic, however, in this chapter the constraint parameters are also allowed to be stochastic. If only the parameter b is stochastic, then handling the constraint is straight forward. Furthermore, if the parameter b is given by a Gaussian distribution then the constraint $H^{\mathrm{T}}\mathbb{X} - b$ is also a Gaussian distribution. Consequently, the methods in Chapter 4 would be directly applicable after calculating the mean and covariance of the constraint.

However, if both parameters (H and b) are stochastic, then the evaluation of the chance constraints as well as the solution of the optimization program becomes more complex. In particular, by allowing H and b to be uncertain, the distribution of $H^{\mathrm{T}}\mathbb{X} - b$ becomes a sum of multiple products of random variables. In general, the properties of the probability distribution for this constraint are not easy to calculate analytically, increasing the complexity of the problem. The following section describes three approaches to deal with such uncertain constraint parameters.

7.2 Chance Constraints with Uncertain Parameters and Variables

This section describes three approaches for dealing with the added complexity in the chance constraints. The first approach, Boole's method, exploits properties of the constraint's probability distribution and approximates it with a distribution that is easier to evaluate. The second approach uses sampling to represent the uncertainty and uses the CVaR method to enforce the chance constraints. In the third approach, a novel hybrid method is developed that uses both analytic functions and sampling to represent the probability distributions.

7.2.1 Boole's Method

In Boole's method, the joint chance constraints are upper bounded by a sum of univariate constraints:

$$\mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} > b\right) \leq \sum_{i=1}^{q} \mathrm{P}(h_i^{\mathrm{T}}\mathbb{X} > b_i). \tag{7.2}$$

Due to the central limit theorem, if the dimension of the state space is large enough then the expression $h_i^{\mathrm{T}}\mathbb{X}_j - b_i$ can be approximated well by a univariate Gaussian distribution, eliminating the need to evaluate multivariate integrals. Given this conversion, the probability of constraint violation can be evaluated efficiently.

If the system state and the constraint parameters are independent, then the mean of the distribution is,

$$E[h_i^{\mathrm{T}}\mathbb{X} - b_i] = E[h_i]^{\mathrm{T}}E[\mathbb{X}] - E[b_i] = \bar{h}_i^{\mathrm{T}}\bar{\mathbb{X}} - \bar{b_i}, \tag{7.3}$$

and the covariance is,

$$\mathrm{var}(h_i^{\mathrm{T}}\mathbb{X} - b_i) = \sum_{j=1}^{nN} \mathrm{var}(h_{ij}\mathbb{X}_j) + \mathrm{var}(b_i) + \sum_{j,k}^{nN} \mathrm{cov}(h_{ij}\mathbb{X}_j, h_{ik}\mathbb{X}_k) + \sum_{j=1}^{nN} \mathrm{cov}(h_{ij}\mathbb{X}_j, b_i). \tag{7.4}$$

To simplify later notation, let $\sigma_i^2 = \mathrm{var}(h_i^{\mathrm{T}}\mathbb{X} - b_i)$.

Recall from Section 3.1.2 that by using Boole's inequality, the multivariate constraints are bounded by a sum of univariate constraints in Eqn. (3.18). Each univariate constraint can be approximated well as a Gaussian with mean and variance as given in Eqns. (7.3) and (7.4) respectively. The univariate constraints can then be efficiently evaluated through

$$\begin{aligned} \mathrm{P}(h_i^{\mathrm{T}}\mathbb{X} - b_i > 0) &= \frac{1}{\sqrt{2\pi}} \int_{\frac{\bar{b}_i - \bar{h}_i^{\mathrm{T}}\bar{\mathbb{X}}}{\sigma_i}}^{\infty} \exp\left(-\frac{z^2}{2}\right) dz \\ &= 1 - \Phi\left(\frac{\bar{b}_i - \bar{h}_i^{\mathrm{T}}\bar{\mathbb{X}}}{\sigma_i}\right), \end{aligned} \tag{7.5}$$

where $\Phi(\cdot)$ is the Gaussian cumulative distribution function. Although this function does not have an analytic solution, it can be efficiently evaluated using a series approximation or a lookup table.

Optimization Program

Program (7.6) shows the resulting optimization problem using Boole's method with uncertain constraint parameters. The program modifies the program (4.37) for deterministic parameters by incorporating the evaluation of the chance constraints in Eqn. (7.5). The variable that is being optimized is the desired trajectory, $\mathbb{X}^d$, for the controller to follow.

$$\begin{aligned} \underset{\mathbb{X}^d}{\text{minimize}} \quad & \mathbf{E}\left[\phi(\mathbb{X}, \mathbb{U})\right] \\ \text{subject to} \quad & \\ & \bar{\mathbb{X}} = T^{xx}\bar{x}_0 + T^{xx^d}\mathbb{X}^d \\ & \bar{\mathbb{U}} = T^{ux_0}\bar{x}_0 + T^{ux^d}\mathbb{X}^d \\ & \bar{\mathbb{U}} \in F_{\mathbb{U}} \\ & z_i = \bar{b}_i - \bar{h}_i^{\mathrm{T}}\bar{\mathbb{X}}, \ \forall i \\ & \sigma_i^2 = \mathrm{var}(h_i^{\mathrm{T}}\mathbb{X} - b_i), \ \forall i \\ & 1 - \Phi(\tfrac{z_i}{\sigma_i}) \le \epsilon_i, \ \forall i \\ & \textstyle\sum_{i=1}^{q} \epsilon_i \le \delta \end{aligned} \tag{7.6}$$

The optimization program (7.6) is not necessarily convex. The convexity of the program depends on the convexity of the chance constraints, which are convex only if $\Phi(\cdot)$ is concave. It can be shown that the function $\Phi(\frac{z_i}{\sigma_i})$ is concave if z_i/σ_i is both positive and concave (from convex composition rules). Since $\bar{b}_i - \bar{h}_i^{\mathrm{T}}\bar{\mathbb{X}}$ preserves the convexity or concavity of $1/\sigma_i$, the concavity of z_i/σ_i depends solely on the concavity of $1/\sigma_i$. Unfortunately, σ_i may be the composition of a concave, non-decreasing function (square root) with a convex expression (quadratic) of the state mean which is not guaranteed to be convex or concave on the entire domain. As a result, the program cannot be guaranteed to be convex.

Gradient of the Chance Constraints

The gradient of the chance constraints can be computed analytically to aid in the solution of the optimization problem. From the chain rule,

$$\nabla_{\mathbb{X}^d}\mathrm{P}\left(h_i^{\mathrm{T}}\mathbb{X} - b_i > 0\right) = \frac{\partial \mathrm{P}\left(h_i^{\mathrm{T}}\mathbb{X} - b_i > 0\right)}{\partial \bar{\mathbb{X}}}\nabla_{x^d}\bar{\mathbb{X}}. \tag{7.7}$$

From Eqn. (7.5) and the Leibniz integral rule,

$$\frac{\partial \mathrm{P}\left(h_i^{\mathrm{T}}\mathbb{X} - b_i > 0\right)}{\partial \bar{\mathbb{X}}} = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(b_i - h_i^{\mathrm{T}}\bar{\mathbb{X}})^2}{2\sigma_i^2}\right)\left[\frac{\bar{h}_i^{\mathrm{T}}}{\sigma_i} + \frac{\bar{b}_i - \bar{h}_i^{\mathrm{T}}\bar{\mathbb{X}}}{\sigma_i^2}\frac{\partial \sigma_i}{\partial \bar{\mathbb{X}}}\right]. \tag{7.8}$$

Finally, the gradient of $\bar{\mathbb{X}}$ with respect to $\mathbb{X}^d$ is $\nabla_{\mathbb{X}^d}\bar{\mathbb{X}} = T^{xx^d}$ which results in,

$$\nabla_{\mathbb{X}^d}\mathrm{P}\left(h_i^{\mathrm{T}}\mathbb{X} - b_i > 0\right) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(b_i - h_i^{\mathrm{T}}\bar{\mathbb{X}})^2}{2\sigma_i^2}\right)T^{xx^d\mathrm{T}}\left[\frac{\bar{h}_i^{\mathrm{T}}}{\sigma_i} + \frac{\bar{b}_i - \bar{h}_i^{\mathrm{T}}\bar{\mathbb{X}}}{\sigma_i^2}\frac{\partial \sigma_i}{\partial \bar{\mathbb{X}}}\right]^{\mathrm{T}}. \tag{7.9}$$

Qualification of the Gaussian Constraint Approximation

The methodology for Boole's approach assumed that the distribution of the constraints can be approximated well by a Gaussian distribution. In order to qualify this approximation, a Monte Carlo simulation was performed to compute the distribution of the constraint. Figure 7.1 shows the resulting histogram of the value of the constraint from the Monte Carlo simulation as compared with the analytical Gaussian approximation (dotted line). As the figure shows, the Gaussian approximation matches the Monte Carlo evaluation of the constraint well.

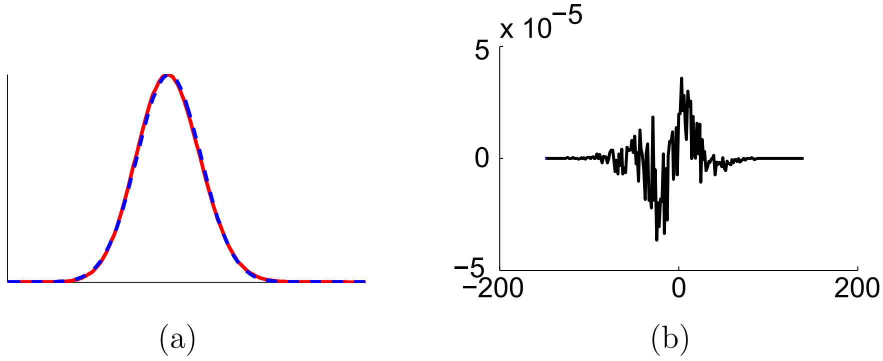


Figure 7.1: A comparison of the analytical Gaussian approximation for the chance constraint with uncertain parameters. (a) Randomly generated constraint (solid) and the Gaussian approximation (dotted). (b) The error of the approximation.

Another Monte Carlo simulation of the Gaussian approximation was performed to compare the predicted probability of constraint violation. The constraint and system parameters were generated such that the true probability of constraint violation was 0.1. The average Gaussian approximation error for this case was 1.64%. The reason the Gaussian approximation overestimates the violation probability is because the tails of the distribution are difficult to capture through random sampling.

7.2.2 CVaR Method

Another approach to handling the uncertainty in the constraint parameters is to use sampling. In this approach, the constraint parameters are sampled in addition to the distributions of the process noise, measurement noise, and initial state. This results in the following set of particles

$$\begin{gathered}\{w_0^{(1)}, \ldots, w_{N-1}^{(1)}, \ldots, w_0^{(N_s)}, \ldots, w_{N-1}^{(N_s)}\},\\ \{v_0^{(1)}, \ldots, v_{N-1}^{(1)}, \ldots, v_0^{(N_s)}, \ldots, v_{N-1}^{(N_s)}\},\\ \{x_0^{(1)}, \ldots, x_0^{(N_s)}\},\\ \{H^{(1)}, \ldots, H^{(N_s)}\},\\ \{b^{(1)}, \ldots, b^{(N_s)}\}.\end{gathered} \tag{7.10}$$

In order to use the CVaR method, the expectation in Eqn. (3.41) is not only over the system state but also over the uncertain constraint parameters. To calculate the expectation, each system trajectory particle $\mathbb{X}^{(j)}$ is associated with a set of constraint parameters $H^{(j)}$ and $b^{(j)}$ for all $j = 1, \ldots, N_s$ as follows

$$\mathrm{E}\left[\left(\max\left(H^{\mathrm{T}}\mathbb{X} - b\right) + \alpha\right)_+\right] \approx \frac{1}{N_s}\sum_{j=1}^{N_s}\left(\max\left(H^{(j)\mathrm{T}}\mathbb{X}^{(j)} - b^{(j)}\right) + \alpha\right)_+. \tag{7.11}$$

7.2.3 Hybrid Method

The two approaches presented thus far for handling uncertain constraint parameters either approximate the constraints' probability distribution analytically or use sampling. However, both approaches have limitations which can be reduced by employing a hybrid approach using a combination of analytical functions and sampling to represent the probability distributions. In particular, for some problem formulations, the stochastic variables are naturally separable into two sets: one set that can be accurately represented through an analytic function, and another set whose distribution is best represented through sampling. This section develops this hybrid method, discusses how to handle the joint chance constraints under the separation of variables, and shows that the resulting optimization program is convex.

Figure 7.2 shows an example of a problem whose stochastic variables naturally separate as described above. In this example, a vehicle is navigating through an uncertain environment. Here, the environment is defined by a set of half plane constraints defined by a series of end points, which are assumed to be normally distributed. The figure shows a solution using an insufficient number of particles (50) to truly represent the underlying distribution of the stochastic variables. This results in 30 times the allowed probability of constraint violation for this solution. The ellipses show the 99.7% confidence regions for the exact distribution of the stochastic variables. As is illustrated, the samples of the system state do not truly represent the underlying distribution, whereas the samples of the uncertain environment represent the wall end point distributions well. Consequently, in this example an improved solution could be found by representing the uncertain environment through sampling and representing the system state through its exact analytical function.

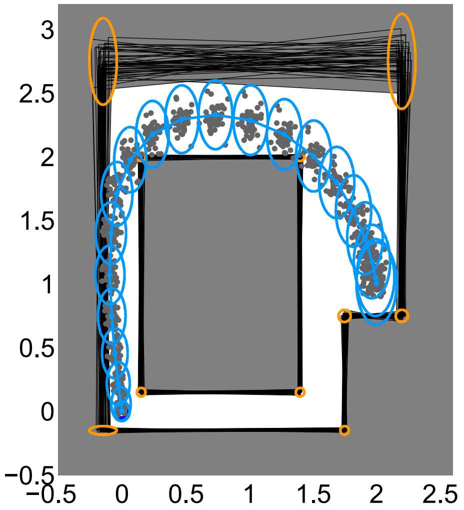


Figure 7.2: An illustration of the solution for a small number of particles (50) used to represent the uncertainty.

In the following development of the hybrid method, sampling is used to represent the uncertain constraints and analytical functions are used to represent the uncertainty of the system state. A similar approach can be used to employ the hybrid method for other separations of the uncertainty representation.

For the chosen uncertainty representation, only the constraint parameters are sampled, resulting in the following set of particles

$$\begin{gathered}\{H^{(1)}, \ldots, H^{(N_s)}\},\\ \{b^{(1)}, \ldots, b^{(N_s)}\}.\end{gathered} \tag{7.12}$$

The distribution of the closed-loop state, $\mathbb{X}$, and the closed-loop control input, $\mathbb{U}$, are represented analytically with distributions given by Eqn. (4.24). Using this hybrid representation of the uncertainty, none of the current methods for enforcing the joint chance constraints, $\mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} - b > 0\right) \leq \delta$, are readily applicable.

Using Boole's method with sampling of the uncertain constraint parameters, the probability of constraint violation is approximated as follows:

$$\begin{gathered}\frac{1}{N_s}\sum_{j=1}^{N_s}\left(1 - \mathrm{P}(h_i^{(j)\mathrm{T}}\mathbb{X} > b_i^{(j)})\right) \leq \epsilon_i, \quad i = 1, \ldots, q, \\ \sum_{i=1}^{q}\epsilon_i \leq \delta.\end{gathered} \tag{7.13}$$

Using the analytical function for the probability distribution of the state, this simplifies even further to

$$\begin{gathered}\frac{1}{N_s}\sum_{j=1}^{N_s}\left(1 - \Phi\left(\frac{b_i^{(j)} - h_i^{(j)\mathrm{T}}\bar{\mathbb{X}}}{\sqrt{h_i^{(j)\mathrm{T}}\Sigma_{\mathbb{X}}h_i^{(j)}}}\right)\right) \leq \epsilon_i, \quad i = 1, \ldots, N_c \\ \sum \epsilon_i \leq \delta.\end{gathered} \tag{7.14}$$

The final optimization program for the hybrid approach is given by:

$$\begin{array}{ll} \underset{\mathbb{X}^d}{\text{minimize}} & \mathbf{E}\left[\phi(\mathbb{X}, \mathbb{U})\right] \\ \text{subject to} & \\ & \bar{\mathbb{X}} = T^{xx}\bar{x}_0 + T^{xx_d}\mathbb{X}^d \\ & \bar{\mathbb{U}} = T^{ux_0}\bar{x}_0 + T^{ux_d}\mathbb{X}^d \\ & \bar{\mathbb{U}} \in F_{\mathbb{U}} \\ & z_{ji} = b_i^{(j)} - h_i^{(j)\mathrm{T}}\bar{\mathbb{X}}, \ \forall i, j \\ & \sigma_{ji}^2 = h_i^{(j)\mathrm{T}}\Sigma_{\mathbb{X}}h_i^{(j)}, \ \forall i, j \\ & \frac{1}{N_s}\sum_{j=1}^{N_s}\left(1 - \Phi(\frac{z_{ji}}{\sigma_{ji}})\right) \leq \epsilon_i, \ \forall i \\ & \sum_{i=1}^{q}\epsilon_i \leq \delta \end{array} \tag{7.15}$$

The optimization program is convex if $b_i^{(j)} \geq h_i^{(j)\mathrm{T}}\bar{\mathbb{X}}$ for all $j = 1, \ldots, N_s$ and $i = 1, \ldots, q$. To show this, consider two feasible solutions $\left(\bar{\mathbb{X}}^{(1)}, \epsilon_i^{(1)}\right)$ and $\left(\bar{\mathbb{X}}^{(2)}, \epsilon_i^{(2)}\right)$.

To show the constraints are convex, it suffices to show that the convex combination of two feasible solutions is feasible. Define the convex combination for $0 \leq \theta \leq 1$ as

$$\left(\bar{\mathbb{X}}^{(*)}, \epsilon_i^{(*)}\right) \triangleq \left(\theta\bar{\mathbb{X}}^{(1)} + (1-\theta)\bar{\mathbb{X}}^{(2)}, \theta\epsilon_i^{(1)} + (1-\theta)\epsilon_i^{(2)}\right).$$

From the concavity of the Gaussian cumulative distribution function $\Phi(z)$ for $z \in [0, \infty)$, the lefthand side of Eqn. (7.14) for $\left(\bar{\mathbb{X}}^{(*)}, \epsilon_i^{(*)}\right)$ can be upper bounded by

$$\begin{aligned}
&\frac{1}{N_s}\textstyle\sum_{j=1}^{N_s}\left(1-\Phi\left(\frac{b_i^{(j)} - {h_i^{(j)}}^{\mathrm{T}}\bar{\mathbb{X}}^{(*)}}{\sqrt{{h_i^{(j)}}^{\mathrm{T}}\Sigma_{\mathbb{X}}h_i^{(j)}}}\right)\right)\\
&\leq \frac{1}{N_s}\textstyle\sum_{j=1}^{N_s}\left(1-\theta\Phi\left(\frac{b_i^{(j)} - {h_i^{(j)}}^{\mathrm{T}}\bar{\mathbb{X}}^{(1)}}{\sqrt{{h_i^{(j)}}^{\mathrm{T}}\Sigma_{\mathbb{X}}h_i^{(j)}}}\right) - (1-\theta)\Phi\left(\frac{b_i^{(j)} - {h_i^{(j)}}^{\mathrm{T}}\bar{\mathbb{X}}^{(2)}}{\sqrt{{h_i^{(j)}}^{\mathrm{T}}\Sigma_{\mathbb{X}}h_i^{(j)}}}\right)\right).
\end{aligned}$$

After simplifying and collecting common terms

$$= 1 - \frac{1}{N_s}\sum_{j=1}^{N_s}\theta\Phi\left(\frac{b_i^{(j)} - {h_i^{(j)}}^{\mathrm{T}}\bar{\mathbb{X}}^{(1)}}{\sqrt{{h_i^{(j)}}^{\mathrm{T}}\Sigma_{\mathbb{X}}h_i^{(j)}}}\right) - \frac{1}{N_s}\sum_{j=1}^{N_s}(1-\theta)\Phi\left(\frac{b_i^{(j)} - {h_i^{(j)}}^{\mathrm{T}}\bar{\mathbb{X}}^{(2)}}{\sqrt{{h_i^{(j)}}^{\mathrm{T}}\Sigma_{\mathbb{X}}h_i^{(j)}}}\right). \quad (7.16)$$

From the definition of the two feasible solutions it is known that

$$-\frac{1}{N_s}\sum_{j=1}^{N_s}\Phi\left(\frac{b_i^{(j)} - {h_i^{(j)}}^{\mathrm{T}}\bar{\mathbb{X}}^{(k)}}{\sqrt{{h_i^{(j)}}^{\mathrm{T}}\Sigma_{\mathbb{X}}h_i^{(j)}}}\right) \leq \epsilon_i^{(k)} - 1 \quad k = 1, 2.$$

Substituting this into Eqn. (7.16) and simplifying yields,

$$\begin{aligned}
&\leq 1 + \theta(\epsilon_i^{(1)} - 1) + (1-\theta)(\epsilon_i^{(2)} - 1)\\
&= \theta\epsilon_i^{(1)} + (1-\theta)\epsilon_i^{(2)}\\
&= \epsilon_i^{(*)}.
\end{aligned}$$

Hence the constraint in Eqn. (7.14) is convex in $\bar{\mathbb{X}}$ and ϵ_i. Therefore, the optimization program (7.15) is also convex if $b_i^{(j)} \geq {h_i^{(j)}}^{\mathrm{T}}\bar{\mathbb{X}}$ for all $j = 1, \ldots, N_s$ and $i = 1, \ldots, q$. Given a general uncertainty model of the constraints, there is no guarantee that

$b_i^{(j)} \geq h_i^{(j)\mathrm{T}} \bar{\mathbb{X}}$. However, if the constraints $b_i^{(j)} - h_i^{(j)\mathrm{T}} \bar{\mathbb{X}} \geq 0 \ \forall i, j$ are added to the optimization program and none of the constraints are active, then the globally optimal solution can be guaranteed.

As will be illustrated in the following examples, the hybrid approach has several benefits over the Gaussian approximation approach and the CVaR method. Since the hybrid approach is a convex program, it typically can be solved faster than the non-convex program in the Gaussian approximation method. Due to using both sampling and analytical distributions to represent the uncertainty, fewer particles are needed to fully represent the underlying uncertainty than when only using sampling. As compared to the CVaR method, this drastically reduces the computational complexity of the problem formulation.

7.3 Case Study: Motion Planning in a 2-Dimensional Environment

The previously described methods for handling uncertain constraints are demonstrated in this section through two examples of stochastic motion planning in a 2D environment.

7.3.1 Environment Description

The environment is assumed to be defined by a set of half plane constraints defined by a series of end points. It is assumed that the end points for the i-th constraint, e_1^i and e_2^i, are independent and normally distributed,

$$e_j^i \sim \mathcal{N}\left(\begin{bmatrix} \bar{e}_{xj}^i \\ \bar{e}_{yj}^i \end{bmatrix}, \begin{bmatrix} \sigma^2_{e_{xj}^i} & 0 \\ 0 & \sigma^2_{e_{yj}^i} \end{bmatrix} \right), \quad \forall i. \tag{7.17}$$

In an abuse of notation, let $p_k = [x_k \ y_k]^\mathrm{T}$ be the 2D position of the vehicle at time-step k with mean $\bar{p}_k = [\bar{x}_k \ \bar{y}_k]^\mathrm{T}$ and covariance $\mathrm{cov}\,(p_k) = \begin{bmatrix} \sigma^2_{x_k} & \sigma^2_{x_k y_k} \\ \sigma^2_{x_k y_k} & \sigma^2_{y_k} \end{bmatrix}$. The expression for the feasible region of the system state is then:

$$F_{\mathbb{X}} = \left\{x : h_i^{\mathrm{T}} p_k \leq b_i, \forall (i,k) \in \mathcal{C}\right\} \tag{7.18}$$

where $\mathcal{C}$ is the set of constraints on the system. The vector h_i is defined as the outward pointing normal of half plane i (and without loss of generality is assumed to be a 90^o counterclockwise rotation of the vector between the end points):

$$h_i = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} e^i_{x2} - e^i_{x1} \\ e^i_{y2} - e^i_{y1} \end{bmatrix} = \begin{bmatrix} e^i_{y1} - e^i_{y2} \\ e^i_{x2} - e^i_{x1} \end{bmatrix}. \tag{7.19}$$

The term b_i can be calculated using the normal vector, h_i, and one of the end points,

$$b_i = h_i^{\mathrm{T}} \begin{bmatrix} e^i_{x2} \\ e^i_{y2} \end{bmatrix} = e^i_{x2} e^i_{y1} - e^i_{x1} e^i_{y2}. \tag{7.20}$$

The variance of the univariate chance constraint is then:

$$\mathrm{var}(h_i^{\mathrm{T}} p_k - b_i) = \alpha_5 \bar{x}_k^2 + \alpha_6 \bar{x}_k + \alpha_7 \bar{y}_k^2 + \alpha_8 \bar{y}_k + \alpha_0, \tag{7.21}$$

and can be calculated through standard probability theory. The parameters α_i can be computed *a priori*, but the variance depends upon the system state and consequently the control inputs, rendering the optimization program (7.6) for the Gaussian approximation method nonconvex.

Given the uncertainty of the end points, the half plane description is distributed via

$$h_i \sim \mathcal{N}\left(\begin{bmatrix} \bar{e}^i_{y1} - \bar{e}^i_{y2} \\ \bar{e}^i_{x2} - \bar{e}^i_{x1} \end{bmatrix}, \begin{bmatrix} \sigma^2_{e^i_{y1}} + \sigma^2_{e^i_{y2}} & 0 \\ 0 & \sigma^2_{e^i_{x1}} + \sigma^2_{e^i_{x2}} \end{bmatrix}\right), \tag{7.22}$$

and the constant b_i has a distribution equal to the difference of two products of Gaussians.

The chance constraint expands to,

$$\mathrm{P}\left(\underbrace{e^i_{y_1}x_k - e^i_{y_2}x_k + e^i_{x_2}y_k - e^i_{x_1}y_k + e^i_{x_1}e^i_{y_2} - e^i_{x_2}e^i_{y_1}}_{c_{ik}} > 0\right). \tag{7.23}$$

7.3.2 Evaluation of the Chance Constraints For Boole's Method

The expression from Eqn. (7.23),

$$c_{ik} = e^i_{y_1}x_k - e^i_{y_2}x_k + e^i_{x_2}y_k - e^i_{x_1}y_k + e^i_{x1}e^i_{y_2} - e^i_{x_2}e^i_{y1} \tag{7.24}$$

is the sum of products distributed by the modified Bessel function of the second kind. Fortunately due to the large number of terms and the central limit theorem, the expression is characterized well by a Gaussian distribution.

From probability theory, the expected value and variance of the product of uncorrelated, univariate Gaussians $x \sim \mathcal{N}(\mu_x, \sigma_x^2)$ and $y \sim \mathcal{N}(\mu_y, \sigma_y^2)$ is,

$$\mathrm{E}\left[xy\right] = \mu_x\mu_y \tag{7.25}$$

$$\mathrm{var}\left(xy\right) = \mu_x\sigma_y^2 + \mu_y\sigma_x^2 + \sigma_x^2\sigma_y^2. \tag{7.26}$$

The mean of Eqn. (7.24) is easily characterized by:

$$\mathrm{E}\left[c_{ik}\right] = \bar{e}^i_{y1}\bar{x}_k - \bar{e}^i_{y_2}\bar{x}_k + \bar{e}^i_{x_2}\bar{y}_k - \bar{e}^i_{x_1}\bar{y}_k + \bar{e}^i_{x_1}\bar{e}^i_{y_2} - \bar{e}^i_{x2}\bar{e}^i_{y_1}, \tag{7.27}$$

but the variance is more difficult to evaluate. Combining all the terms and simplifying yields,

$$\mathrm{var}\left(c_{ik}\right) = \underbrace{\alpha_1\sigma^2_{x_k} + \alpha_2\sigma^2_{y_k} + \alpha_3\sigma^2_{x_k y_k} + \alpha_4}_{\alpha_0} + \alpha_5\bar{x}_k^2 + \alpha_6\bar{x}_k + \alpha_7\bar{y}_k^2 + \alpha_8\bar{y}_k \tag{7.28}$$

where

$$\begin{aligned}
\alpha_1 = &\ (\bar{e}^i_{y_1})^2 + \sigma^2_{e^i_{y_1}} + (\bar{e}^i_{y_2})^2 + \sigma^2_{e^i_{y_2}} - 2\bar{e}^i_{y_1}\bar{e}^i_{y_2}, \\
\alpha_2 = &\ (\bar{e}^i_{x_1})^2 + \sigma^2_{e^i_{x_1}} + (\bar{e}^i_{x_2})^2 + \sigma^2_{e^i_{x_2}} - 2\bar{e}^i_{x_1}\bar{e}^i_{x_2}, \\
\alpha_3 = &\ 2\bar{e}^i_{y_1}\bar{e}^i_{x_2} - 2\bar{e}^i_{y1}\bar{e}^i_{x_1} - 2\bar{e}^i_{y_2}\bar{e}^i_{x_2} + 2\bar{e}^i_{y_2}\bar{e}^i_{x_1}, \\
\alpha_4 = &\ (\bar{e}^i_{x_1})^2\sigma^2_{e^i_{y_2}} + (\bar{e}^i_{y2})^2\sigma^2_{e^i_{x_1}} + \sigma^2_{e^i_{y_2}}\sigma^2_{e^i_{x_1}} + (\bar{e}^i_{x_2})^2\sigma^2_{e^i_{y_1}} + (\bar{e}^i_{y_1})^2\sigma^2_{e^i_{x_2}} + \sigma^2_{e^i_{y_1}}\sigma^2_{e^i_{x_2}}, \\
\alpha_5 = &\ \sigma^2_{e^i_{y_1}} + \sigma^2_{e^i_{y_2}}, \\
\alpha_6 = &\ \bar{e}^i_{x_2}\sigma^2_{e^i_{y_1}} + \bar{e}^i_{x_1}\sigma^2_{e^i_{y_2}}, \\
\alpha_7 = &\ \sigma^2_{e^i_{x_1}} + \sigma^2_{e^i_{x_2}}, \\
\alpha_8 = &\ \bar{e}^i_{y_2}\sigma^2_{e^i_{x_1}} + \bar{e}^i_{y_1}\sigma^2_{e^i_{x_2}}.
\end{aligned}$$

The variance in Eqn. (7.28) can be separated into two parts: term α_0 which can be computed *a priori*, and a set of terms that depend upon the system state and consequently the control inputs. The final form of the variance is,

$$\mathrm{var}(c_{ik}) = \alpha_5\bar{x}_k^2 + \alpha_6\bar{x}_k + \alpha_7\bar{y}_k^2 + \alpha_8\bar{y}_k + \alpha_0. \tag{7.29}$$

7.3.3 Uncertain Constraint Parameter Examples

In the following examples, the three proposed solution methods (Boole's, CVaR, and hybrid) are used to plan trajectories through two sample environments. Both environments contain nonconvex regions which can be transformed into a series of convex problems as in [86], [87]. The approach taken in the following examples was to decompose the feasible region into convex tunnels which are then planned through.

Example 8. *The following example is similar to Example 4 except for a* $Q_{\mathrm{obj}} = 50I$ *and* $\delta = 0.005$. *Here* δ *was decreased to highlight the differences between the proposed methods.*

The parameters for the points of the environment are,

$$\begin{aligned}
e_1 &= [-0.15, -0.15], & \Sigma_{e_1} &= \mathrm{diag}(0.001, 0.0001), \\
e_2 &= [-0.15, 2.75], & \Sigma_{e_2} &= \mathrm{diag}(0.001, 0.01), \\
e_3 &= [2.2, 2.75], & \Sigma_{e_3} &= \mathrm{diag}(0.001, 0.012), \\
e_4 &= [2.2, 0.75], & \Sigma_{e_4} &= \mathrm{diag}(0.0002, 0.0002), \\
e_5 &= [1.75, 0.75], & \Sigma_{e_5} &= \mathrm{diag}(0.0002, 0.0002), \\
e_6 &= [1.75, -0.15], & \Sigma_{e_6} &= \mathrm{diag}(0.0001, 0.0001), \\
e_7 &= [0.15, 0.15], & \Sigma_{e_7} &= \mathrm{diag}(0.0001, 0.0001), \\
e_8 &= [0.15, 2], & \Sigma_{e_8} &= \mathrm{diag}(0.0001, 0.0001), \\
e_9 &= [1.4, 2], & \Sigma_{e_9} &= \mathrm{diag}(0.0001, 0.0001), \\
e_{10} &= [1.4, 0.15], & \Sigma_{e_{10}} &= \mathrm{diag}(0.0001, 0.0001).
\end{aligned} \tag{7.30}$$

The solution for Example 8 is shown in Figure 7.3(a)-(c) for Boole's method, the CVaR method, and the hybrid method, respectively. The blue, solid line is the trajectory of the system when accounting for the uncertainty of the environment, and the green, dotted line is the solution when planning only through the mean environment. The blue ellipses show the 99.7% confidence ellipsoid of the system state at each time-step. For this example, the optimal path is always through the top region, even though the bottom region is shorter. The bottom region is infeasible with respect to the chosen allowed constraint violation because of the large uncertainty of the vertical position of the state.

There are several interesting differences between the solution that account for the uncertainty and the solution through the mean environment. The solution from the mean environment initially curves toward the wall with large uncertainty, but when the uncertainty of the environment is incorporated the system deviates away from it. The more noticeable difference between the two solutions in the top region of the environment. The solution which accounts for the uncertainty of the environment stays lower to avoid the highly uncertain top wall.

The approximate solution using Boole's method takes 61.2 seconds to compute, the CVaR method uses 1500 particles and takes 286.1 seconds to solve, and the hybrid approach uses only 50 particles for the environment and takes 2.42 seconds. Clearly, using the hybrid approach would enable stochastic control in real-time applications, whereas the other two approaches could only be used for offline calculations. The estimated true probability of constraint violation using Monte Carlo simulation for Boole's method, the CVaR method, and the hybrid method is 0.0038, 0.0047, and 0.0042. The CVaR method results in the least conservative solution and the approximate solution using Boole's method has the most conservative solution. Even though the hybrid method yields a more conservative solution over the CVaR approach, this is outweighed by the significant reduction in computation time.

Table 7.1 shows the simulated constraint violation for $\delta = 0.01$ and $\delta = 0.005$ when planning with and without the uncertainty of the environment using Boole's method. The solution which only uses the mean environment significantly violates the allowed constraint violation, while the solution which accounts for the uncertainty of the environment is conservative due to the approximation from Boole's inequality. In particular, Boole's inequality doesn't account for the dependence between the state at different time-steps violating the constraints. Consider a wall whose uncertainty is purely translational. If at one time-step the system doesn't violate the constraint, then the system will also not violate the constraint for any future motion parallel to the wall. Using Boole's inequality to bound the probability ignores this dependency. The conservativeness can be reduced, at the expense of more computation, by using one of Bonferroni's inequalities [95].

Table 7.1: A comparison of the constraint violation for Boole's method using different environmental models.

Sol. Method	δ	δ_{sim}
Mean Env.	0.005	0.0123
Env. Uncertainty	0.005	0.0038
Mean Env.	0.01	0.017
Env. Uncertainty	0.01	0.0088

Table 7.2 compares the statistics of the probability of constraint violation using the CVaR method with various numbers of particles. The statistics were calculated using 100 runs and an allowed probability of constraint violation of $\delta = 0.005$. For

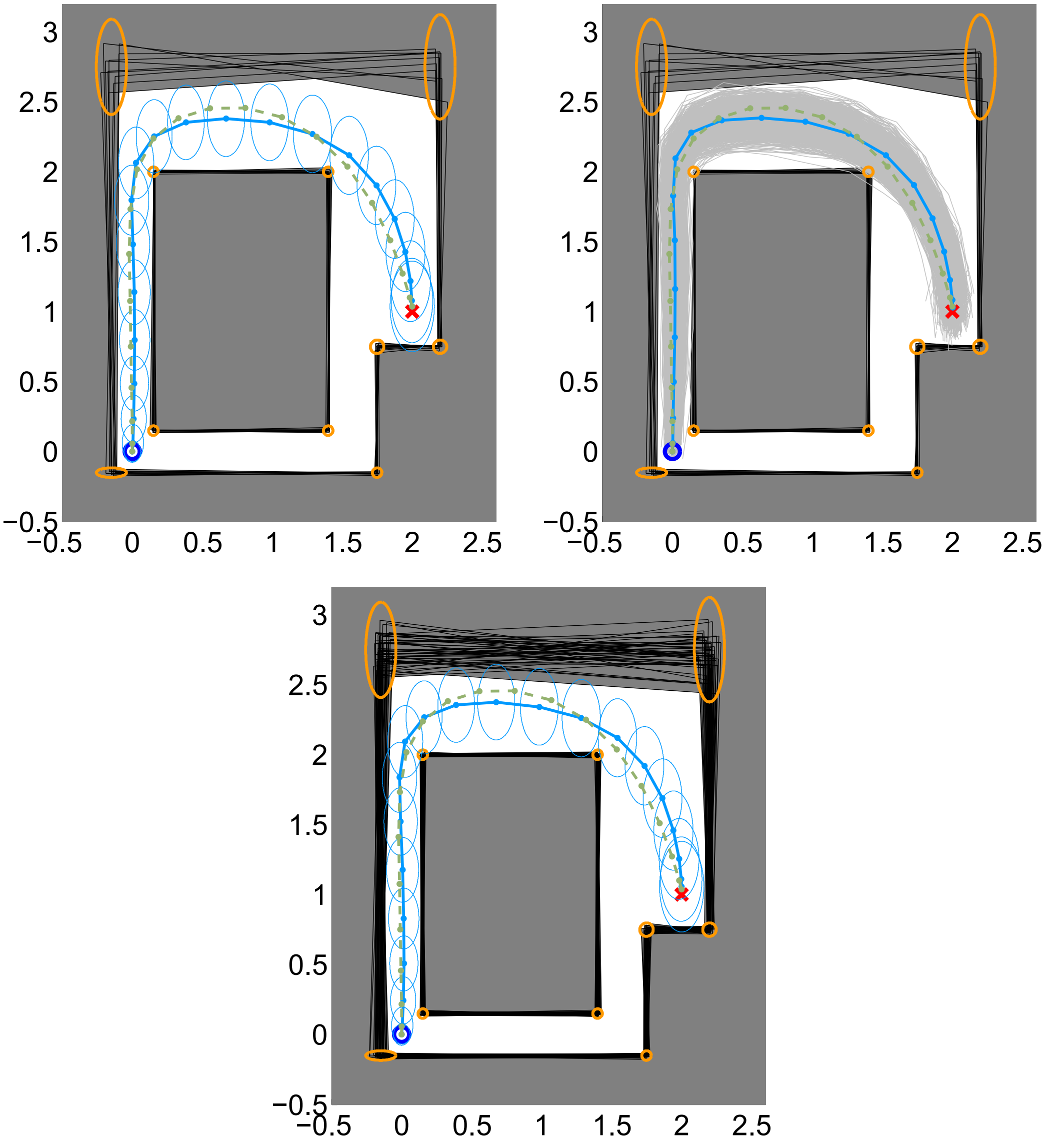


Figure 7.3: Results for Example 8 with ten realizations of the uncertain walls. The white area is the feasible region of the system state and the orange ellipses are the uncertainty of the environment. The blue, solid line is the solution when accounting for the uncertainty of the environment and the green, dotted line plans through the mean environment. The blue ellipses around the path indicate the uncertainty of the system. The start and goal location are marked by an 'o' and 'x', respectively. (a) The approximate solution using Boole's method. (b) The solution using the CVaR method with 1500 particles and their trajectories are shown as the gray lines. (c) The solution using the hybrid method with $N_s = 50$ samples.

the smallest number of particles, the CVaR method on average violates the allowed violation probability by 2.5 times. With 1500 particles, on average the constraint violation is below the user allowed amount, but the maximum is twice the allowed amount.

Table 7.2: A comparison of the solution statistics for the CVaR algorithm for various number of particles for the uncertain constraint parameter problem. $\delta = 0.005$.

N_s	mean δ_{sim}	std. dev. δ_{sim}	max δ_{sim}
500	0.0127	0.0050	0.027
1000	0.0070	0.0026	0.014
1500	0.0048	0.0017	0.010

A comparison of the solution statistics for the hybrid method is shown in Table 7.3 for an example with $\delta = 0.005$. Even for a very small number of particles, $N_s = 25$, the mean probability of constraint violation is equal to the allowed amount. As the number of particles increases, the standard deviation and maximum probability of constraint violation decreases while the conservativeness of the solution increases.

Table 7.3: A comparison of the solution statistics for the hybrid method for various number of particles for the uncertain constraint parameter problem.

N_s	mean δ_{sim}	std. dev. δ_{sim}	max δ_{sim}	mean comp. time (secs)
10	0.0058	0.0014	0.0098	0.95
25	0.0050	0.0010	0.0078	1.48
50	0.0046	0.0009	0.0075	2.42
100	0.0045	0.0007	0.0062	4.56
200	0.0044	0.0006	0.0063	6.53
300	0.0043	0.0005	0.0054	9.11
400	0.0041	0.0005	0.0053	11.18
500	0.0042	0.0004	0.0051	14.63

Example 9. *The following example is similar to Example 8 except the environment is larger and the noise parameters are changed to,*

$$\begin{aligned} \Sigma_w &= \mathrm{diag}(0.001, 0.001, 0.001, 0.001) \\ \Sigma_v &= \mathrm{diag}(0.002, 0.002). \end{aligned} \tag{7.31}$$

For this example, the solution is shown in Figure 7.4. The solution which accounts for the uncertainty of the environment is shown as the blue, solid line and has a planning horizon of $N = 55$. The green, dotted line plans through the mean environment and uses a shorter horizon of $N = 35$ due to the shorter route. This mean environment solution has a simulated constraint violation of 0.0084, 68% over the allowed violation. When accounting for the uncertainty of the environment, the system cannot take the direct path as the mean environment solution can because the walls in that corridor have too much uncertainty (resulting in a violation of the probability constraint).

The solutions from Boole's method and the hybrid method are shown in Figure 7.4(a)(b), respectively. The violation probability is 0.0038 for Boole's method and 0.0050 for the hybrid method with only 100 particles. The solution using Boole's method is over 13 times slower than the hybrid method.

7.4 Conclusions

This chapter extended the stochastic control framework to handle uncertain constraint parameters. Three different methods were proposed to handle the resulting joint chance constraints. In the first method, the probability distribution was accurately approximated by a Gaussian distribution with an analytic mean and covariance. Unfortunately, the resulting optimization program was nonconvex increasing the computation times and only guaranteeing a locally optimal solution. Inherently, the CVaR method can handle the increased complexity of the uncertain constraint parameters through sampling. Even though the method easily handles the evaluation and satisfaction of the joint chance constraints, the computational complexity of this approach limits its applicability. In this chapter, a novel hybrid approach was proposed that uses both analytical functions and sampling to represent the uncertainty. This method was shown to be superior to the individual CVaR and Boole's method approaches. The hybrid approach resulted in a convex optimization program under certain conditions, guaranteeing the optimal solution. It also decreased the

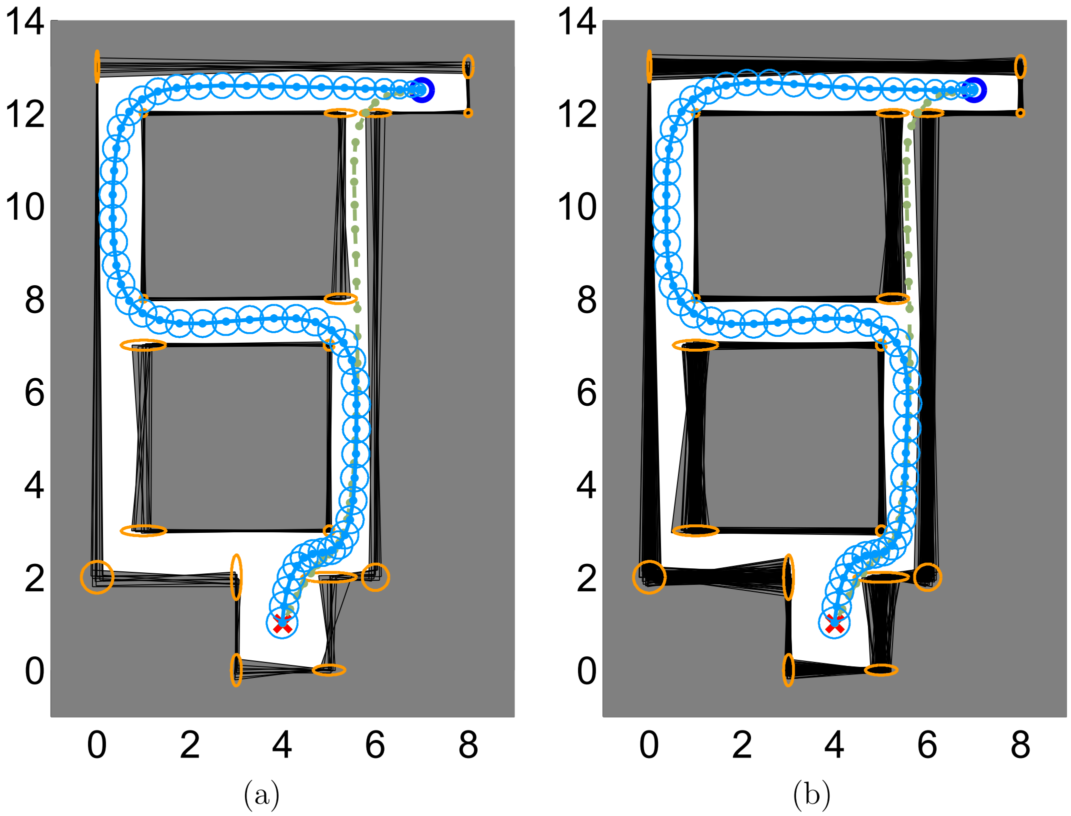


Figure 7.4: The solution for the uncertain constraint parameter for Example 9. (a) The solution using the approximate approach with Boole's method. (b) The hybrid approach with $N_s = 100$ particles for the environment.

conservativeness of the solution over the approximation method. As compared to the CVaR method, typically fewer samples were needed which reduced the computational complexity and could thus enable real-time operation.

Chapter 8

A Hierarchical Method for Stochastic Control with Uncertain Constraint Parameters

In the previous chapter, a stochastic control problem was formulated for handling uncertain constraints. In this chapter, the problem formulation is further extended to incorporate new possible measurements of the uncertain constraint parameters, leading to increased performance of the overall system. This problem is the standard dual control problem with the following two competing objectives,

- Action: To control the system based on the current belief of the state to achieve a desired goal;
- Probing: To explore with the system to acquire more information about its state to enable better control.

Consequently, the system must trade off between completing its objective and exploring to gain more information to better complete its objective.

One way to solve this problem is to calculate a policy for the entire belief space (the space of probability distributions of all possible system states) that determines the best action for every possible future belief state that it may encounter. This requires the system to maintain a probability distribution over all possible system states and constraint parameters. In general, this formulation is intractable due to its large computational complexity arising from the necessity of planning over long time horizons and the dimension of the belief space.

In this chapter, a novel hierarchical method is proposed to reduce the computational complexity by splitting the problem into two stages: (1) exploration and (2) execution. In the exploration phase, the dynamics of the system are abstracted in order to simplify the problem and a high level control policy is calculated that incorporates information from potential new measurements. In the next phase, the system executes the high level control policy by solving a stochastic control problem given the current information of the system state and constraint parameter values. As the system evolves, it continuously incorporates sensor information into its belief of the state and re-solves both stages. An experimental demonstration of the proposed hierarchical method applied to stochastic motion planning for a quadrotor aerial vehicle in a dynamic, uncertain environment is presented in the following chapter.

8.1 Motivating Example for Dual Control

The following example motivates the need for incorporating sensing of the uncertain environment into the stochastic control solution.

Example 10. *The following example is similar to Example 8 except for* $N = 65$ *and* $\delta = 0.05$*. Here the* δ *was chosen to highlight the need to incorporate constraint sensing.*

The solution without incorporating constraint sensing is shown in Figure 8.1. Here, the system must take the longer route around the top of the obstacle because the bottom route violates the chance constraints due to the large environmental uncertainty in the second corridor. This results in a path length of 2 times larger than if the system was able to take the bottom route through the environment.

For this example, the system is forced to take the longer path because environmental sensing is not taken into account. In this case, it might have been beneficial for the system to gain information by exploring the environment to improve the solution of its task. For instance, the shorter path might have been enabled through sensing the walls in the environment while executing its trajectory. The proposed hierarchical framework developed in the following section systematically trades off between exploring to gain information and accomplishing the task while ensuring the chance constraints. This new framework will be applied to this example at the end of the chapter.

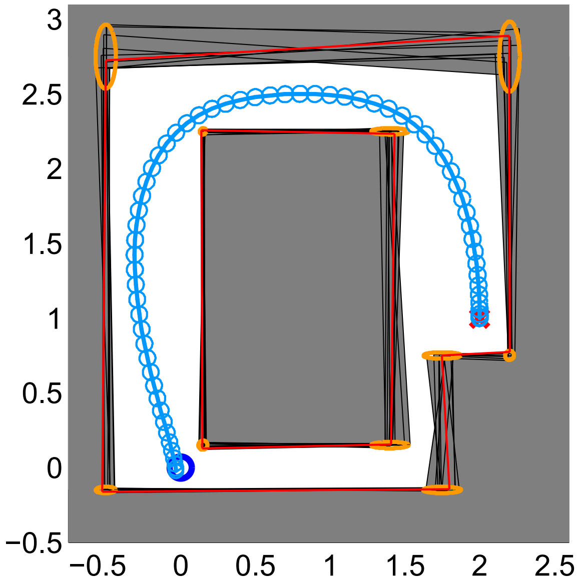


Figure 8.1: The white area is the feasible region of the system state and the orange ellipses are the uncertainty of the environment. The blue, solid line is the solution when accounting for the uncertainty of the environment. The blue ellipses around the path indicate the uncertainty of the system. The start and goal location are marked by an 'o' and 'x', respectively.

8.2 Hierarchical Method

A general principled way of modeling the problem of stochastic motion planning through uncertain environments is by using Partially Observable Markov Decision Processes (POMDPs). A POMDP models a system's decision process in which the system's dynamics are not necessarily deterministic, but rather the outcome of its actions could be stochastic, and the system does not know the true underlying state. Instead, the system must take measurements of the underlying state and maintain a probability distribution over all possible states. In order to solve a POMDP, a policy must be calculated for the entire belief space that determines the best action for every possible future belief state that it may encounter.

The main difficulty in solving a POMDP is its large computational complexity, and in general solving a POMDP exactly is intractable. The two main causes of the large computational complexity is the curse of dimensionality and the long planning horizons. The first difficulty arises from the need to represent the probability distribution over *all* possible states in order to incorporate all possible future events. For the specific problem of stochastic motion planning through uncertain environments, the belief state would be all possible system states as well as all possible realizations of the uncertain environment. The second difficulty is due to the need to use long planning horizons to incorporate information gain resulting in good, non-myopic policies.

In order to overcome the high computational complexity of solving the POMDP formulation of the problem, the problem is broken down into two phases: (i) an expected shortest path problem on an uncertain graph and (ii) a chance constrained optimization problem. By decomposing the problem in this way, the size of the belief space is drastically reduced by planning only where it is best to observe the environment. This in turn greatly reduces the computational complexity. The method is shown in Algorithm 4.

The first step of the algorithm decomposes the environment into a set of nodes and edges as shown in Figure 8.2. Then the probability that the link will be traversable after sensing is calculated. After this graph is built, the expected shortest path problem is solved to provide a high level plan for the chance constrained stochastic motion planning algorithm presented in Section 7.2.3. Then the system executes the plan and senses the environment to update the probability distribution of the environment. This procedure repeats until the system completes its objective. The individual steps of the algorithm are described in more detail below.

Algorithm 4 Hierarchical Stochastic Motion Planning Algorithm

```
while objective not complete do
    Decompose the environment into a graph of nodes and edges
    Calculate the probabilities of the edges being traversable after sensing
    Solve the expected shortest path problem
    Solve the chance constrained stochastic motion planning problem
    Execute the plan
    Sense the environment
end while
```

8.2.1 Environmental Decomposition

The first step in using the hierarchical stochastic motion planning algorithm is to decompose the environment into a graph. One approach to perform this decomposition is to use a weighted medial axis decomposition [96] based upon the uncertainty of the environment. Another approach would be to build a graph based upon motion primitives. Care should be taken to provide a decomposition of the environment that is as dynamically feasible as possible to ensure that the system can execute the high level plan, and that the cost of traversing the edges is as accurate as possible. Once the graph has been constructed, the probability that each link is traversable needs to be calculated. In order to account for new environmental information, the probability associated with each link is the *a posterior* probability, rather than the *a priori*, as this allows the system to determine whether there is information to be gained by exploring a certain area.

To simplify the discussion, the environmental uncertainty is assumed to only be in the parameter b. The probability of collision before sensing for a single constraint (i.e. $H^{\mathrm{T}}\mathbb{X} \in \mathbb{R}$ and $b \in \mathbb{R}$) is calculated via

$$\mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} > b\right) = \int\int \mathbf{1}\left(H^{\mathrm{T}}\mathbb{X} > b\right)\mathrm{P}(b)\mathrm{P}(\mathbb{X})db d\mathbb{X}. \tag{8.1}$$

However, the statistic needed for each link in the expected shortest path problem is the probability that *after* sensing the link will be traversable. In order to calculate this, an allowed probability of collision (ϵ) for the constraint needs to be allocated which will be discussed in the following section. Once the allowed probability of collision

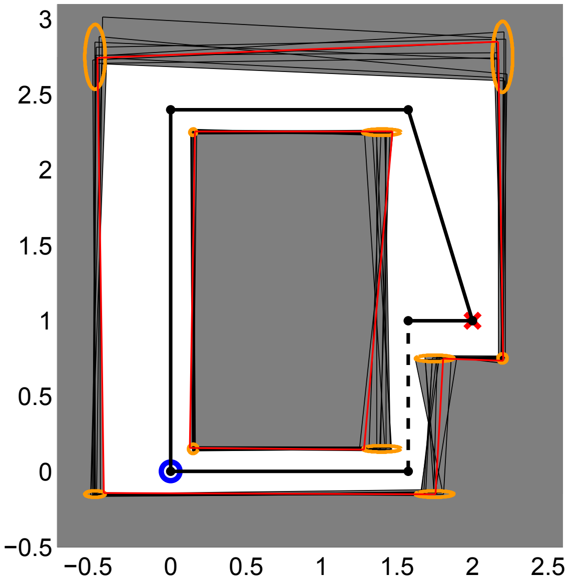


Figure 8.2: A graph based decomposition of the environment. The dotted link has a large probability of failure before sensing but after sensing it has a large probability of being traversable.

for the constraint has been determined, the probability the link will be traversable after sensing is calculated via

$$\mathrm{P}\left(\mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} > b|H, b\right) \leq \epsilon\right), \tag{8.2}$$

where ϵ is the allowed probability of constraint violation associated with the constraint. For the current derivation with uncertainty only in the parameter b, Eqn. (8.2) can be simplified further to

$$\mathrm{P}\left(\mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} > b|b\right) \leq \epsilon\right) = \int \mathbf{1}\left(\mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} > b|b\right) \leq \epsilon\right)\mathrm{P}(b)db. \tag{8.3}$$

If b^* can be found such that

$$\begin{aligned} \mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} > b|b\right) &\leq \epsilon, \quad b \geq b^* \\ \mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} > b|b\right) &> \epsilon, \quad b < b^* \end{aligned}$$

then the probability in Eqn. (8.3) can be simplified to

$$\mathrm{P}\left(\mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} > b|b\right) \leq \epsilon\right) = \int_{b^*}^{\infty} \mathrm{P}(b)db. \tag{8.4}$$

For a given link, if there is more than one constraint, then a similar methodology can be used to determine an analytic formula for the probability.

If both parameters H and b are uncertain, the complexity of calculating the required probability increases because it requires evaluating multivariate integrals, however, numerical integration through Monte Carlo sampling can be used to efficiently evaluate it.

Consider now how this new formulation would effect the results in the previous example shown in Figure 8.1. The probability of failure for traversing the dotted link in Figure 8.2 *before* sensing is 0.15, but the probability that the link will be traversable *after* sensing is 0.64 for an allowed probability of failure of $\epsilon = 0.02$. The large probability of failure before sensing requires the system to take the long route around the obstacle in Figure 8.1. However, the large probability that the link will be traversable *after* sensing suggests that it might be beneficial to first investigate whether or not the corridor is traversable and then either proceed directly to the goal or backtrack around the top of the obstacle.

The best strategy for navigating the environment can be calculated by solving the expected shortest path problem through the defined uncertain graph, however the strategy is highly impacted by the risk assigned to each edge. The following section discusses a method for allocating these risks through convex optimization.

8.2.2 Risk Allocation for the Graph Decomposition

To calculate the probability of failure after sensing, the allowed risk for each link needs to be known. The allowed risks have a direct impact on the routes generated by the expected shortest path algorithm. If the allowed risk is improperly assigned, then it may cause certain paths that should be explored to be overlooked. Therefore it is necessary to carefully allocate the risk in order to correctly trade off between exploration and execution in the expected shortest path calculation. The method presented below is a heuristic based upon solving a convex approximation of the original problem.

An example environment decomposition is given by the undirected graph in Figure 8.3. Here, p_{ij} is the probability after sensing that the edge between node i and j will be passable and is defined as $p_{ij} = f_{ij}(\epsilon_{ij})$, with $f_{ij}(\epsilon_{ij})$ given by Eqn. (8.2). Let e_{ij} be the edge between node i and j. For any path through the graph, there is a set of edges $\{e^{(1)}, \ldots, e^{(n)}\}$ with associated probabilities $\{p^{(1)}, \ldots, p^{(n)}\}$ and probability functions $\{f^{(1)}(\epsilon^{(1)}), \ldots, f^{(n)}(\epsilon^{(n)})\}$.

In general the probability functions f_{ij} are positive, nondecreasing functions of the allowed risk, but they are neither convex nor concave for the entire domain. One example of the probability function versus the allowed risk is shown in Figure 8.4; for this example the function is log-concave. For cases where the function is not log-concave, a convex approximation of the problem will be obtained by approximating the probability functions with log-concave functions $\hat{f}_{ij}$. In the following discussion, this modified problem formulation will be used to allocate the risk.

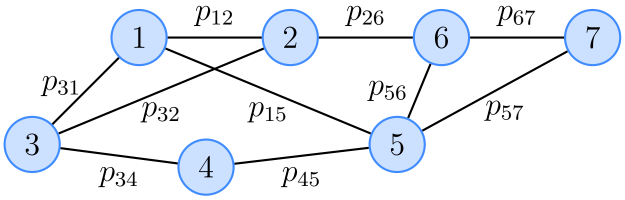


Figure 8.3: An example graph used in the calculation of the expected shortest path.

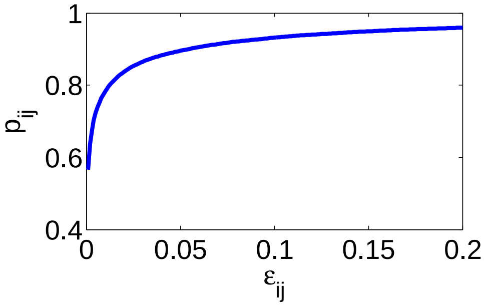


Figure 8.4: An example of the probability of the link being traversable after sensing versus the allowed risk.

One method of assigning the risk for each edge is by maximizing the probability the path will be passable. The *a priori* probability that after sensing the path between any two nodes will be passable is the product of the probabilities for each edge that is traveled, i.e. $\prod_{m=1}^{n} p^{(m)}$. This assignment of risk can be obtained by solving the following optimization program.

$$\begin{array}{ll} \text{maximize} & \prod_{m=1}^{n} f^{(m)}(\epsilon^{(m)}) \\ \text{subject to} & \\ & \epsilon^{(m)} \geq 0, \quad m = 1, \ldots, n \\ & \sum_{m=1}^{n} \epsilon^{(m)} \leq \delta, \quad m = 1, \ldots, n \end{array} \tag{8.5}$$

The optimization program maximizes the probability of success subject to the constraint that the sum of the assigned risk over the path is smaller than the user allowed amount. Unfortunately, this program is not necessarily convex because the objective function is not guaranteed to be concave. To form a convex approximation, the original probability functions can be replaced by their concave approximations, but this still doesn't guarantee convexity. To form a convex optimization program, a set of slack variables are introduced based upon the logarithm of the approximate probability functions and the objective function is replaced by its logarithm. This results in the following equivalent convex optimization program.

$$\begin{array}{ll} \text{maximize} & \sum_{m=1}^{n} \alpha_m \\ \text{subject to} & \\ & \epsilon^{(m)} \geq 0, \quad m = 1, \ldots, n \\ & \sum_{m=1}^{n} \epsilon^{(m)} \leq \delta \\ & \alpha_m \leq \log\left(\hat{f}^{(m)}(\epsilon^{(m)})\right), \quad m = 1, \ldots, n \end{array} \tag{8.6}$$

In order for this new optimization program to be convex, the objective function needs to be concave and the constraints have to be convex. The objective function is a sum of linear variables which is concave and convex. The first two constraints are linear, and the last constraint is convex since it is a linear expression less than a concave function because the functions $\hat{f}^{(i)}$ are log-concave. Given the convexity of the optimization program, it provides an efficient method for allocating the risk for a large number of links.

In this formulation, the optimization program will need to be solved for all paths without cycles between the start and goal location, and it may assign a different allowed risk for each link depending on the path. To obtain a risk assignment for all edges that is independent of the path, the problem formulation can be changed to a single optimization program that maximizes the probability of successfully traveling over *all* paths subject to the same constraints for all paths.

Using the risk allocation computed from the above formulation, the expected shortest path problem is solved next in order to determine the trade off between exploring to gain more information and completing the objective of the planning process.

8.2.3 Expected Shortest Path

The expected shortest path problem on an uncertain graph has been extensively studied. The approach taken in this work is to convert the problem into a Markov Decision Process (MDP) [97]. A MDP is represented by a 4-tuple $(S, A, P(\cdot), R(\cdot))$ where

- S is the set of states
- A is the set of actions that can be taken at each state
- $P_a(s, s')$ is the probability that executing action a in state s will lead to state s'
- $R_a(s, s')$ is the reward function.

Let the graph associated with Figure 8.2 be represented by G with the edges noted as $E(G)$, the vertices as $V(G)$, the probability associated with each edge in the graph, $e \in E(G)$, be represented as $\mathrm{P}(e)$, and the start and end vertices of edge e be e_s and e_e, respectively.

To define the expected shortest path problem on a graph as a MDP, the 4-tuple needs to be defined. For the stochastic motion planning problem, a state, s, includes both the current location of the system as well as its belief of the current state of the environment, i.e. the probabilities associated with all the links in the graph being traversable. The set of all possible states is therefore

$$S = V(G) \times \prod_{e \in E(G)} \{0, 1, \mathrm{P}(e)\}. \tag{8.7}$$

The set of actions is defined as traversing over any edge in the graph, i.e. $A = E(G)$. Next, the transition probability function needs to be defined which must take into account the information gain from executing actions in the uncertain environment. For example, if the system attempts to take edge e, then with probability $\mathrm{P}(e)$ it will reach e_e and with probability $1 - \mathrm{P}(e)$ it will remain at e_s. After taking edge e the probability distribution over the environment will have changed; if the system successfully crosses the edge then $\mathrm{P}(e) = 1$, but if the system remains at e_s then the edge is not traversable and the probability updates to $\mathrm{P}(e) = 0$. The transition function can be defined as

$$P_a(s, s') = \begin{cases} \mathrm{P}(a) & \begin{array}{l} \text{if } s_p \in a, \\ s'_p \in a, \\ s_p \neq s' \\ s'(a) = 1; \end{array} \\ 1 - \mathrm{P}(a) & \begin{array}{l} \text{if } s_p = s'_p \\ s_p \in a, \\ s'(a) = 0; \end{array} \\ 0 & \text{otherwise} \end{cases} \tag{8.8}$$

where an action a is an associated edge e, $s(a)$ is the probability of the action a being traversable in state s, and s_p is the location of the system in state s. Lastly, the reward function for the problem is,

$$R_a(s, s') = \begin{cases} \gamma & \text{if s = g} \\ c(s, s') & \text{otherwise} \end{cases} \tag{8.9}$$

where g is the goal location of the planning problem, $\gamma \in \mathbb{R}_+$ is the reward of reaching the goal location, and $c(s, s') \in \mathbb{R}$ is the cost of traveling between state s and s'. The cost function $c(\cdot)$ should include any potential costs, such as energy consumption or path length, that the system would encounter for attempting to navigate that route.

There are many different solution methodologies to solve for the exact solution of a MDP, such as value or policy iteration. However, the formulation of the expected shortest path problem as a MDP has an exponential growth of states with respect to the number of uncertain edges in the graph. Since each uncertain edge can have three values associated with it, $\{0, 1, \mathrm{P}(e)\}$, the number of states in the graph could be as large as $|V(G)| \cdot 3^{|E(G)|}$. This exponential growth of the number of states will prohibit solving for the exact solution for large sized problems. Fortunately, there are many approximate MDP solvers [98] that can handle large size problems and provide near optimal solutions for the exploration phase of the algorithm.

8.2.4 Execution of Expected Shortest Path Solution

Once the expected shortest path problem has been solved to determine the route the system should take, the standard stochastic motion planning problem can be formulated and solved as shown in Chapter 7. Since parts of the uncertain environment are too risky to pass through before sensing, the stochastic motion planning algorithm only navigates the system to intermediate waypoints in the environment. These intermediate waypoints act as decision points for the system to gather more information about the environment to determine if the environment is safe to pass through. As the system is navigating to the intermediate waypoint, it continually senses the environment, updates its probability distribution, determines the next intermediate waypoint from the expected shortest path plan, and then solves another stochastic motion planning problem to reach the next intermediate waypoint.

8.3 Example using Hierarchical Dual Control Method

Example 10 is solved through the proposed hierarchical stochastic motion planning algorithm and the solution is shown in Figure 8.5. The hybrid method in Section 7.2.3 is used to solve the chance constrained stochastic motion planning algorithm, but any of the other methods could have been used. After solving the expected shortest path problem, the best path for the system to take is around the bottom of the obstacle, even though the probability of failure in the second corridor before sensing violates the constraints. Consequently, the system only executes a subset of the plan to the decision point as shown in Figure 8.5(a). After it reaches this waypoint, it senses the environment and updates the probability distribution of the environment, and determines that for this environment the corridor is passable. Consequently, it solves another chance constrained stochastic motion planning problem to proceed directly to the intended goal location as shown in Figure 8.5(b). However, if after sensing the environment the system determined that the corridor was indeed impassable then it would have backtracked and proceeded around the top of the obstacle as displayed in Figure 8.5(c).

8.4 Conclusions

This chapter extended the stochastic control framework to incorporate new information gain from sensing the uncertain constraint parameters to increase the performance of the system. To accomplish this, the system must trade-off between completing its objective and exploring to gain more information to better control the system. In general, the problem is intractable due to its large computational complexity arising from the necessity of planning over long time horizons and the dimension of the belief space.

A novel hierarchical method was developed that reduced the computational complexity. It accomplished this by splitting the problem into two stages: (i) an exploration phase that uses an abstraction of the dynamics to calculate the high level

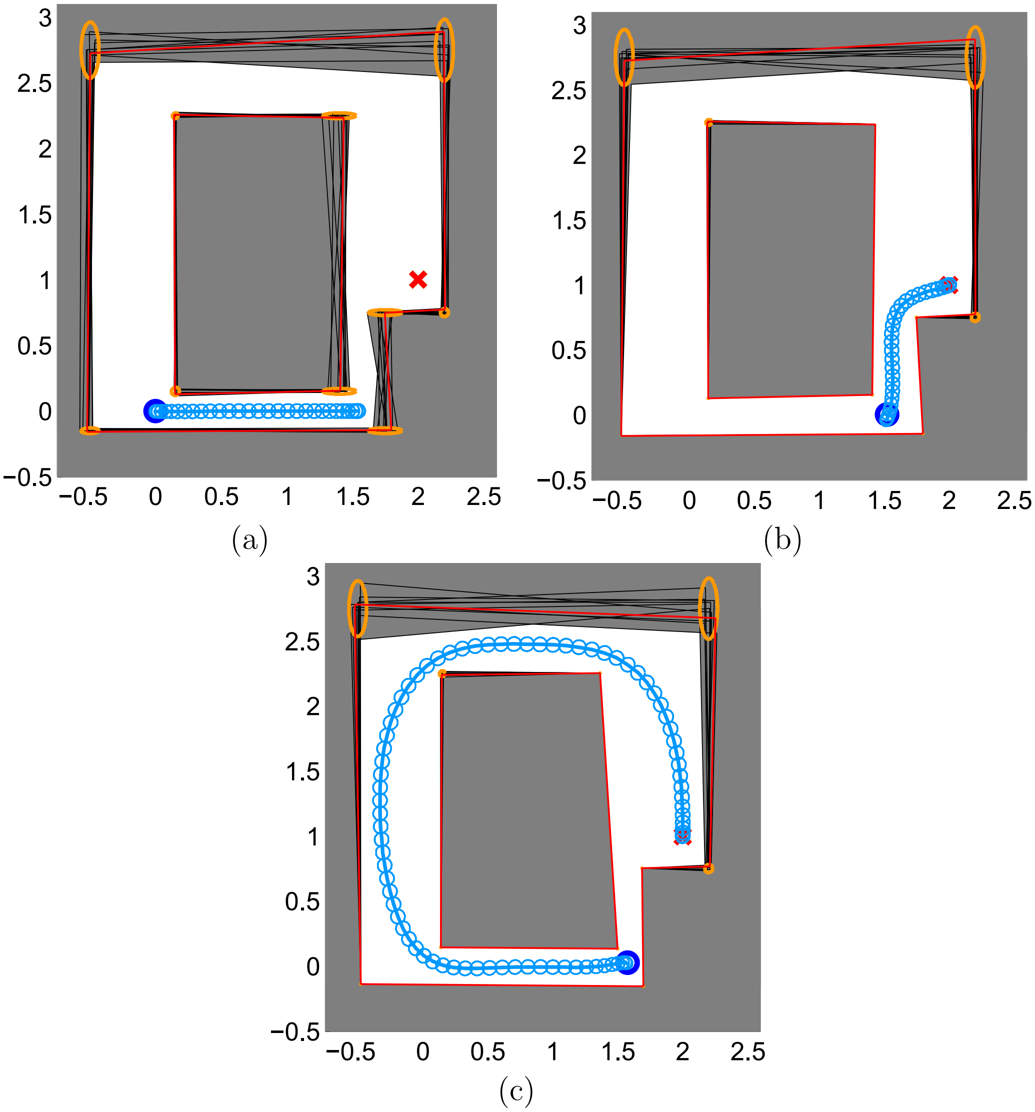


Figure 8.5: The solution from the hierarchical stochastic motion planning algorithm for Example 10. (a) The expected shortest path algorithm determined the best strategy for this environment was to attempt to take the shortest path to the goal location to be able to sense the environment to determine if the route was feasible. (b) After the measurements of the environment have been incorporated into the model of the environment, the system replans the path which results in being able to take the shortest path to the goal location. (c) The stochastic motion planning solution from the sensing point for another realization of the environment. After sensing the environment, the probability of failure is too large which requires the system to backtrack and take the long route around the top of the obstacle.

policy that incorporates information from potential new measurements, (ii) an execution phase that solves a stochastic control problem to follow the high level plan given the current information of the belief space.

The following chapter presents an experimental demonstration of the hierarchical method for a quadrotor navigating through an uncertain 3D environment.

Chapter 9

Experimental Demonstration

The hierarchical stochastic motion planning algorithm presented in the previous chapter was evaluated on a quadrotor unmanned aerial vehicle pictured in Figure 9.1. The vehicle has an onboard inertial measurement unit which provides three-axis attitude, attitude rate and acceleration measurements. An external Vicon positioning system is also used to provide measurements of the quadrotor's position with respect to a global coordinate frame. To sense the environment, the quadrotor is equipped with a Kinect sensor that provides a dense 3D point cloud measurement of the environment.

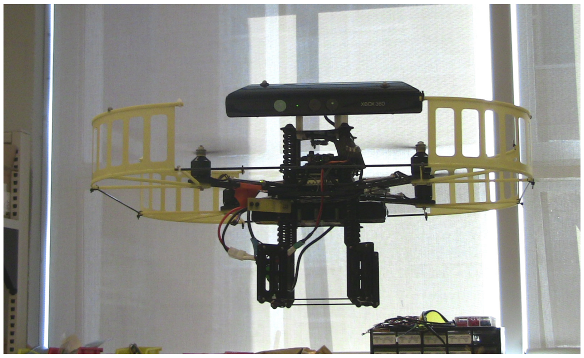


Figure 9.1: An image of the Ascending Technologies pelican quadrotor with the attached Kinect sensor.

9.1 Quadrotor Dynamics

A high level illustration of how the quadrotor controls its dynamics is shown in Figure 9.2. Mechanically, the quadrotor is much simpler than a traditional helicopter; its blades are fixed pitch, which means that the thrust from each motor is constrained to the vertical axis of the motor. The quadrotor has four rotors consisting of two counter-rotating pairs, which helps to cancel the yaw torque. To increase or decrease the total thrust, the speed of all four rotors are either increased or decreased (which maintains the quadrotor at its current attitude). To change the roll or pitch of the quadrotor, the speed of one rotor is increased while the corresponding rotor is decreased; this keeps the yaw torque and total thrust constant. To affect the yaw of the quadrotor, the speed of one pair of rotors is increased and the other pair is decreased to keep the total thrust constant. Lateral acceleration can be achieved by varying the roll and/or pitch of the quadrotor.

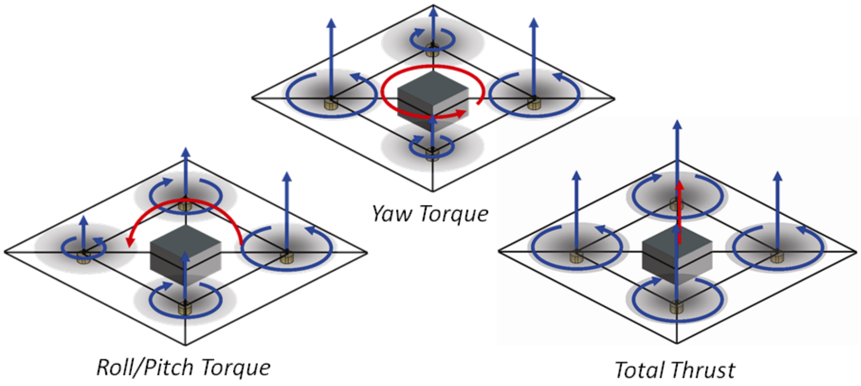


Figure 9.2: An illustration of how the controls affect the quadrotor's dynamics.

The dynamics of the quadrotor are nonlinear, and for simplicity, the aerodynamic effects caused by the changing airflow will be ignored. A free body diagram of the quadrotor is shown in Figure 9.3. The body frame of the quadrotor is attached to the center of mass. The following rotation matrix is used to convert from the body

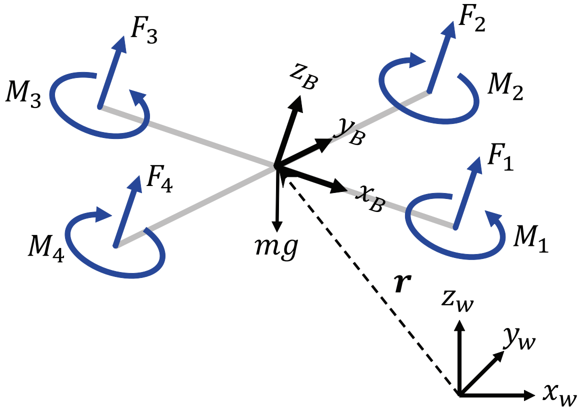


Figure 9.3: A free body diagram for the quadrotor unmanned vehicle.

frame to the world frame

$$R = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix}, \tag{9.1}$$

where $c\theta$ and $s\theta$ are $\cos(\theta)$ and $\sin(\theta)$, and ϕ, θ and ψ are the roll, pitch and yaw Euler angles. From the standard rigid body equations of motion, the translational dynamics of the quadrotor are given by:

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{4} F_i \end{bmatrix}, \tag{9.2}$$

where g is the acceleration of gravity, m is the mass of the quadrotor, and F_i is the force that each motor generates. A similar procedure can be followed for analyzing the rotational dynamics. The angular acceleration in the body frame p, q, r can be determined by applying the standard Euler equations

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} l(F_2 - F_4) \\ l(F_3 - F_4) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{9.3}$$

where $I = \text{diag}(I_{xx}, I_{yy}, I_{zz})$ is the inertia matrix for the quadrotor, l is the length of each rotor arm, and M_i is the torque generated by each motor.

Combining common terms of Eqn. (9.3) yields

$$\begin{bmatrix} I_{xx}\dot{p} \\ I_{yy}\dot{q} \\ I_{zz}\dot{r} \end{bmatrix} = \begin{bmatrix} l(F_2 - F_4) - qr(I_{zz} - I_{yy}) \\ l(F_3 - F_4) - pr(I_{xx} - I_{zz}) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix}. \tag{9.4}$$

Typically, the quadrotor operates around a nominal hover condition with small attitude angles ($\pm 20^o$), and under these conditions the dynamics are represented well by a linearized system. Without loss of generality, assume the yaw angle is held constant at 0 radians. Assuming small angles for roll and pitch and dropping higher order terms, the nonlinear translational dynamics simplify to

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} \theta \sum_{i=1}^{4} F_i \\ -\phi \sum_{i=1}^{4} F_i \\ -mg + \sum_{i=1}^{4} F_i \end{bmatrix}, \tag{9.5}$$

verifying that for small angles the translational acceleration is linear in the attitude of the quadrotor.

In typical operating conditions, the yaw velocity, r, is small, and near nominal hovering conditions $\dot{\phi} = p$, $\dot{\theta} = q$,$\dot{\psi} = r$. Consequently, after dropping higher order terms, the linearized rotational dynamics simplify to

$$\begin{bmatrix} I_{xx}\ddot{\phi} \\ I_{yy}\ddot{\theta} \\ I_{zz}\ddot{\psi} \end{bmatrix} = \begin{bmatrix} l(F_2 - F_4) \\ l(F_3 - F_4) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix}, \tag{9.6}$$

which is linear in the rotor forces which are proportional to the speed of the rotors. This also validates the use of a linear model around the nominal hover condition of the quadrotor dynamics.

9.2 Ascending Technologies Pelican Quadrotor Dynamics

Direct control of the rotor speeds is not possible for the Ascending Technologies quadrotor, however a low level controller is provided that accepts desired roll, pitch, and yaw angles as input. Consequently, the transfer functions from desired attitude angles to attitude and translational dynamics had to be determined through standard system identification techniques [99]. The quadrotor state and control input are

$$x_k = \begin{bmatrix} x & y & z & v_x & v_y & v_z & \theta & \phi & v_\theta & v_\phi \end{bmatrix}^{\mathrm{T}}, \quad u_k = \begin{bmatrix} \phi_d & \theta_d & u_z \end{bmatrix}^{\mathrm{T}}. \tag{9.7}$$

The dynamics are discretized with a time-step of $\Delta t = 0.1$ seconds and the linear system is

$$x_{k+1} = Ax_k + Bu_k + w_k, \tag{9.8}$$

where

$$A = \begin{bmatrix} A_p & A_{pa} \\ 0 & A_a \end{bmatrix}, \tag{9.9}$$

with

$$A_p = \begin{bmatrix} 1 & 0 & 0 & 0.10 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0.10 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0.10 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{9.10}$$

$$A_{pa} = \begin{bmatrix} 0.05 & 0 & 0 & 0 \\ 0 & 0.05 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.98 & 0 & 0 & 0 \\ 0 & 0.98 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad A_a = \begin{bmatrix} 0.23 & 0 & 0.06 & 0 \\ 0 & 0.23 & 0 & 0.06 \\ -4.81 & 0 & 0.71 & 0 \\ 0 & -4.81 & 0 & 0.71 \end{bmatrix}, \tag{9.11}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0.15 & 0 & 3.57 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.15 & 0 & 3.57 \\ 0 & 0 & 0.005 & 0 & 0 & 0.10 & 0 & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}}. \tag{9.12}$$

9.3 Stochastic Motion Planning Algorithm

The procedure for planning safe trajectories for the quadrotor through general environments is described in Algorithm 5. It is assumed that an initial map of the environment is given. With this map, the world is discretized into cells, with each cell storing the distance to the closest obstacle. Given the current start and goal location, the expected shortest path problem is solved to determine the high level plan of how the quadrotor should attempt to reach the goal location. This step trades-off the cost and benefit of exploring riskier, shorter paths. Once the intermediate goal location is determined, the method described in [100] is used to determine a safe pre-path through the environment as input for the chance constrained optimization algorithm. The pre-path method forms a network of free space bubbles from the discretized grid and uses A^* to find the shortest safe path to the intermediate goal location. An example of the pre-path method is shown in Figure 9.4. The start and goal locations are the green and red spheres, respectively. The shortest path is the red line with the network of free space bubbles shown as the blue transparent spheres. The environment around this pre-path is decomposed into a set of convex polytopes. Using the resulting tunnel, the chance constrained optimization program is solved for the trajectory that satisfies the allowed probability of collision constraints. Finally, the quadrotor executes this trajectory, continuously sensing the environment and updating its model of the world.

The point cloud library [101] is used to process the Kinect point cloud data, and the Kinect sensor is operating at 30 Hz. The environment is represented as a set of 3D points corresponding to obstacles. For every frame of the Kinect sensor, points are added to the environment if it sees a new obstacle or removed if an obstacle is no longer there. If at any point the quadrotor determines that the high level route determined from the expected shortest path problem will violate the allowed

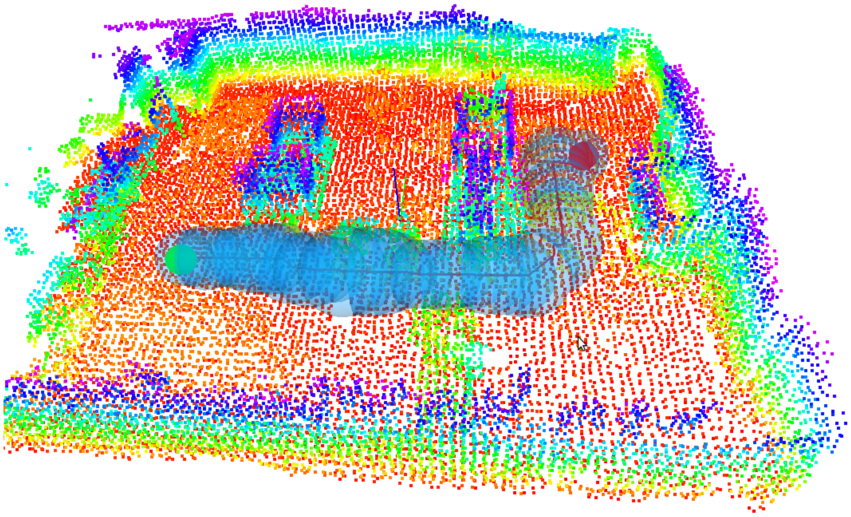


Figure 9.4: An illustration of the pre-path method planning with the network of bubbles around the shortest path. The start is the green sphere, the goal is the red sphere, and the transparent spheres is the network of free space bubbles around the shortest path.

probability of collision, it will re-iterate through Algorithm 5.

Algorithm 5 Stochastic Motion Planning Algorithm

```
Discretize the world
For each grid cell determine the closest obstacle
Solve the expected shortest path problem to determine the intermediate goal location
Perform $A^*$ search from the start to the goal to determine a safe pre-path
Generate a tunnel of convex polytopes around the pre-path
Solve the stochastic motion planning problem
Execute the trajectory, continuously sensing and updating the world model
```

9.4 Results

The hierarchical stochastic motion planning algorithm presented in Chapters 8 and 9 was solved in real-time on the quadrotor testbed. The experimental environment is shown in Figure 9.5. The room has a size of $7.6m \times 7.1m \times 2m$, and is cluttered with desks, chairs, boxes, saw horses, and cabinets that force the quadrotor to use the entire 3-dimensional volume in successfully reaching its goal.

In this example there are two sources of environment uncertainty: (i) the obstacle locations are not known exactly due to the sensing error of the Kinect sensor, and (ii) the cabinet that is outlined with the orange dotted line in Figure 9.5 has a 0.3 probability of its door being open. If the door is open, the quadrotor is prevented from taking the shortest path to the goal location. The error from the Kinect sensor can be directly incorporated into the chance constrained stochastic motion planning algorithm by including it in the probability distribution of the uncertain environment. The uncertainty of the door will be incorporated into the probability of the link through the corresponding corridor.

For planning purposes, the dynamics are discretized with a time-step of $\Delta t = 0.1$ seconds and the resulting plan is sub-sampled to be executed at 20 Hz. An initial map of the environment with the door closed was constructed by manually flying the quadrotor around.

The proposed hierarchical motion planning algorithm was then solved in real-time with the door open. The results are shown in Figure 9.6 (each point represents a part of the environment and is color coded based upon the height off the floor). The solution from the expected shortest path problem instructed the quadrotor to first determine if the cabinet door was open, then either proceed directly to the goal location or backtrack through the middle corridor. As the quadrotor was executing the motion plan, it sensed that the cabinet door was open, and re-solved the stochastic planning problem. Figure 9.6 shows the planned trajectory and executed trajectory as the orange and blue lines, respectively.

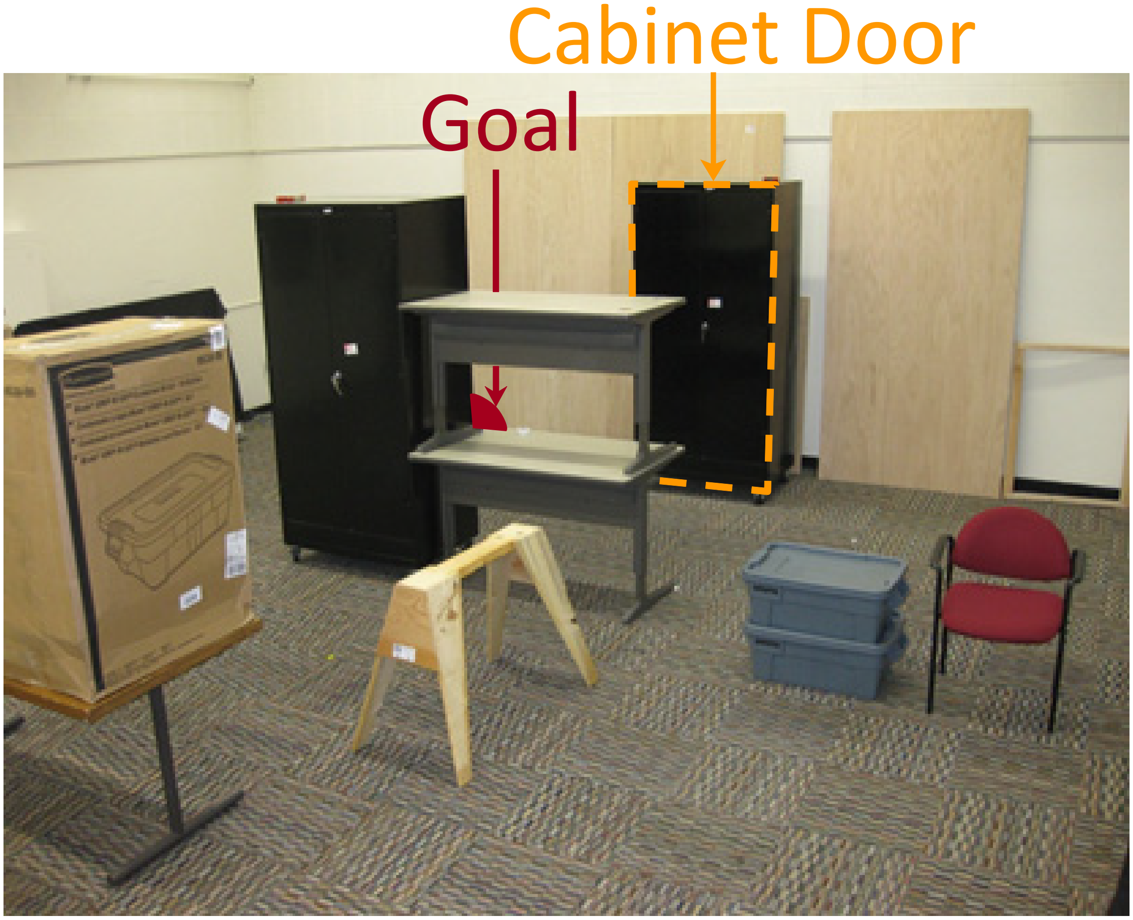


Figure 9.5: An image of the stochastic environment. The start position is out of view on the bottom lefthand corner of the image and the goal location is marked by the red circle which is behind the large cabinet. The cabinet on the right of the image marked by the orange dotted rectangle is stochastic, as the door could either be open or closed which might prevent the robot from passing through that corridor.

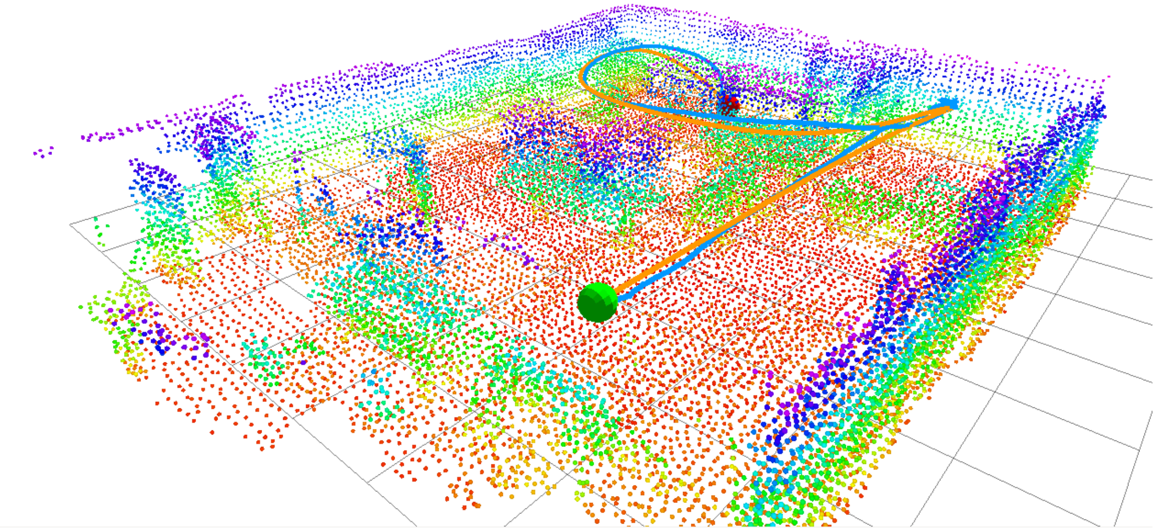


Figure 9.6: The final point cloud representation of the environment. The start location is the green sphere in the lower lefthand corner and the goal location is the red sphere in the upper lefthand corner. The points correspond to obstacle locations and are color coded based upon the height off the floor. The planned trajectory is shown as the orange line through the environment and the trajectory of the quadrotor is shown as the blue line.

Chapter 10

Conclusions and Future Research Directions

This thesis presented stochastic control methods to enable robust operation of autonomous systems under uncertainty. This chapter briefly reviews the contributions of this thesis and discusses potential areas for future research.

10.1 Review of Contributions

The primary goal of this thesis was to increase the applicability of chance constrained programming methods. This was accomplished by extending the framework to include optimization of the feedback controller, predictive control of systems with component failures, constraint parameter uncertainty, and incorporation of future constraint parameter sensing.

Since the feedback controller can shape the uncertainty of the system, incorporating it into the optimization can improve the ability to satisfy the stochastic constraints. Prior work used the ellipsoidal relaxation method to satisfy the joint chance constraints, however, this resulted in a large amount of conservatism degrading the performance of the overall system. In this thesis, a novel method was developed to optimize over both the risk allocation and the feedback controller, and was shown to enable previously infeasible solutions.

This thesis also presented methods which improved the applicability of stochastic control for systems with component failures. This problem is difficult to solve due to the multi-modal probability distribution. Previous work considered using the mixed integer programming method to solve this problem, but this approach has a large computational complexity limiting its applicability. This thesis proposed applying the CVaR method which drastically reduced the computational complexity, leading to the potential solution of larger problems.

Another important area that was considered in this thesis was the extension of the chance constrained framework to handle problems with both uncertain constraint parameters and variables. This extension presented several challenges. The constraints become sums of products of random variables, increasing the difficulty of evaluating the chance constraints. Either a Gaussian approximation or the CVaR sampling method could be used to represent the constraints, but this is shown to result in either a nonconvex optimization program or increased computational complexity. In addition, to incorporate measurements of the uncertain constraint parameters into the problem formulation, a policy must be calculated for the entire belief space. In general this problem is intractable due to its large computational complexity.

To handle these difficulties two novel algorithms was proposed. A hybrid approach was developed to solve the stochastic control problem with uncertain constraint parameters. This method uses a combination of sampling and analytic functions to represent the probability distributions of the system and constraint parameters. This approach results in a convex optimization program under certain conditions, guaranteeing the optimal solution and reducing the computational complexity. Also, a hierarchical method was developed to incorporate future measurements of the constraint parameters. This method splits the stochastic control problem into two stages: (i) an exploration stage that calculates a high level policy for incorporating information from potential future measurements, (ii) an execution stage that solves a stochastic control problem to execute the high level policy.

Lastly, this thesis presented the first successful, real-time experimental demonstration of chance constrained control with uncertain constraint parameters and variables. The proposed stochastic control methods were solved in real-time to plan trajectories for a quadrotor unmanned aerial vehicle navigating in a three-dimensional cluttered, uncertain environment. These methods enabled the quadrotor to successfully trade-off exploring to gather more information and completing its objective.

10.2 Future Research Directions

The research presented in this thesis can be extended in several interesting areas. This section describes several of these extensions.

State Dependent Noise

For the methods presented in this thesis, the noise for the system state was allowed to be time-varying but the distribution was assumed to be known *a priori*. However, in many cases, the noise parameters are dependent upon the system state. For example, for a ground vehicle, the surface properties depend upon the location of the vehicle relative to the terrain (e.g., the vehicle will have different dynamics on gravel versus sand).

Directly incorporating state dependent noise into the current stochastic control framework results in a nonconvex optimization program. One potential approach for handling state dependent noise is to employ an iterative scheme by locally optimizing around a nominal trajectory. The noise properties would be calculated based upon this nominal trajectory.

An alternative approach, similar to the dual control method for uncertain constraint parameters, is to solve the problem in a two-stage scheme. In the first stage, an abstraction of the dynamics are used to calculate a high-level route. In the second stage, the noise parameters are computed based on the high-level route and are used to formulate the standard stochastic control problem.

Distributed Optimization

This thesis considered the use of sampling methods such as CVaR for solving many of the addressed stochastic control problems. While the sampling methods are attractive for solving these types of problems, their computational complexity grows with the number of samples needed to accurately represent the probability distribution of the system. For some applications, even though the CVaR method results in a convex optimization program, the required number of samples may be prohibitive for real-time application.

In order to increase the real-time applicability of the CVaR method, algorithms for parallelizing the solution of the optimization program should be investigated which can benefit from the rise of multi-core processors. One potential approach is to divide the samples into smaller groups which can be solved faster on separate processors. Given this split into smaller problems, an iterative scheme may be necessary to ensure consensus between the solutions.

Model Predictive Control

There are two exciting extensions in the area of incorporating chance constraints in Model Predictive Control (MPC). MPC is an iterative control algorithm with the following three steps: (1) solve an optimization program for the control inputs, (2) apply the first control input, and (3) update the system state based upon the measurements. This iterative procedure has been very successful in controlling a wide range of systems.

MPC has been applied to solve the stochastic control problem with individual chance constraints. One interesting area of future work is to extend the MPC framework to handle joint chance constraints. This is difficult due to the need to iteratively allocate risk over the future time-steps. If this is not done carefully then the resulting solution might be overly risky, or the solution may be conservative degrading the system performance.

A second potential extension is to improve the application of the sampling method in MPC. Due to the iterative nature of the MPC framework, the previous iteration's solution may be used to enable a more targeted sampling strategy of the uncertainty distributions. This can improve the solution quality and reduce the computational complexity.

Feedback Policy of the Uncertain Constraint Parameters

In the presented solution methodology for handling uncertain constraint parameters, as the probability distribution for the parameters changes the entire optimization program needs to be resolved to take advantage of the new information. To eliminate this, future research could investigate solving for a feedback policy that is not only dependent upon the system state but also on the constraint parameters. This would increase the computational complexity of the optimization program, however the program could be solved offline. This feedback policy of the uncertain constraint parameters would reduce the complexity of the online controller by eliminating the need to resolve the optimization program when the distribution changes.

Bibliography

[1] D. Gerhardus, "Robot-assisted surgery: the future is here," *Journal of Healthcare Management*, vol. 48, no. 4, pp. 242–251, 2003.

[2] R. Alterovitz, K. Goldberg, and A. Okamura, "Planning for steerable bevel-tip needle insertion through 2D soft tissue with obstacles," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Barcelona, Spain), April 2005.

[3] J. M. Romano, R. J. Webster III, and A. M. Okamura, "Teleoperation of steerable needles," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Roma, Italy), April 2007.

[4] R. J. Webster III, J. Memisevic, and A. M. Okamura, "Design considerations for robotic needle steering," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Barcelona, Spain), April 2005.

[5] Worldwatch Institute, *Vital Signs 2007-2008: The Trends That Are Shaping Our Future.* New York, NY, USA: W. W. Norton & Company, 2007.

[6] J. McQuade, "A system approach to high performance buildings," tech. rep., United Technologies Group, April 2009. `http://gop.science.house.gov/Media/hearings/energy09/april28/mcquade.pdf`.

[7] United States Department of Energy, "Buildings energy data book." `http://buildingsdatabook.eren.doe.gov/`, 2009.

[8] F. Oldewurtel, D. Gyalistras, M. Gwerder, C. Jones, A. Parisio, V. Stauch, B. Lehmann, and M. Morari, "Increasing energy efficiency in building climate control using weather forecasts and model predictive control," in *10th REHVA World Congress Clima*, (Antalya, Turkey), 2010.

[9] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium* (G. Lakemeyer and B. Nebel, eds.), Morgan Kaufmann, 2002.

[10] R. Vaughan, N. Sumpter, J. Henderson, A. Frost, and S. Cameron, "Experiments in automatic flock control," *Robotics and Autonomous Systems*, vol. 31, no. 1-2, pp. 109 – 117, 2000.

[11] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge, England: Cambridge University Press, 2004.

[12] A. Charnes, W. W. Cooper, and G. H. Symonds, "Cost horizons and certainty equivalents: An approach to stochastic programming of heating oil," *Management Science*, vol. 4, no. 3, pp. 235–263, 1958.

[13] B. L. Miller and H. M. Wagner, "Chance constrained programming with joint constraints," *Operations Research*, vol. 13, no. 6, pp. 930–945, 1965.

[14] A. Prékopa, "On probabilistic constrained programming," in *Proceedings of the Princeton Symposium on Mathematical Programming*, pp. 113–138, Princeton University Press, 1970.

[15] C. van de Panne and W. Popp, "Minimum-cost cattle feed under probabilistic protein constraints," *Management Science*, vol. 9, no. 3, pp. 405–430, 1963.

[16] S. Kataoka, "A stochastic programming model," *Econometrica*, vol. 31, pp. 181–196, 1963.

[17] C. Lagoa, X. Li, and M. Sznaier, "Application of probabilistically constrained linear programs to risk-adjusted controller design," in *Proceedings of the American Control Conference*, vol. 2, pp. 738–743, 2001.

[18] A. Prékopa, “Probabilistic programming,” in *Stochastic Programming* (A. Ruszczyński and A. Shapiro, eds.), vol. 10 of *Handbooks in Operations Research and Management Science*, pp. 267–351, Elsevier, 2003.

[19] R. Henrion, “Structural properties of linear probabilistic constraints,” *Optimization*, vol. 56, no. 4, pp. 425–440, 2007.

[20] A. Ben-Tal and A. Nemirovski, “Robust solutions of linear programming problems contaminated with uncertain data,” *Mathematical Programming*, vol. 88, pp. 411–424, 2000.

[21] R. T. Rockafellar and S. P. Uryasev, “Optimization of conditional value-at-risk,” *Journal of Risk*, vol. 2, pp. 21–41, 2000.

[22] A. Nemirovski and A. Shapiro, “Convex approximations of chance constrained programs,” *SIAM Journal of Optimization*, vol. 17, no. 4, pp. 969–996, 2006.

[23] L. J. Hong, Y. Yang, and L. Zhang, “Sequential convex approximations to joint chance constrained programs: A Monte Carlo approach,” *Operations Research*, vol. 59, no. 3, pp. 617–630, 2011.

[24] D. van Hessem, C. Scherer, and O. Bosgra, “LMI-based closed-loop economic optimization of stochastic process operation under state and input constraints,” in *Proceedings of the 40th IEEE Conference on Decision and Control*, (Orlando, FL), December 2001.

[25] D. van Hessem and O. Bosgra, “Closed-loop stochastic dynamic process optimization under input and state constraints,” in *Proceedings of the 2002 American Control Conference*, (Anchorage, Alaska), May 2002.

[26] D. van Hessem and O. Bosgra, “A full solution to the constrained stochastic closed-loop MPC problem via state and innovations feedback and its receding horizon implementation,” in *Proceedings of the 42nd IEEE Conference on Decision and Control*, (Maui, HI), December 2003.

[27] D. van Hessem and O. Bosgra, "Closed-loop stochastic model predictive control in a receding horizon implementation on a continuous polymerization reactor example," in *Proceedings of the American Control Conference*, (Boston, MA), June 2004.

[28] D. van Hessem, *Stochastic inequality constrained closed-loop model predictive control with application to chemical process operation.* PhD thesis, Delft University of Technology, 2004.

[29] L. Blackmore, "Robust path planning and feedback design under stochastic uncertainty," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, (Honolulu, HI), August 2008.

[30] M. Shin and J. Primbs, "A fast algorithm for stochastic model predictive control with probabilistic constraints," in *Proceedings of the American Control Conference*, (Baltimore, Maryland), pp. 5489–5494, July 2010.

[31] L. Blackmore and M. Ono, "Convex chance constrained predictive control without sampling," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, (Chicago, Illinois), August 2009.

[32] A. Prékopa, "The use of discrete moment bounds in probabilistic constrained stochastic programming models," *Annals of Operations Research*, vol. 85, pp. 21–38, 1999.

[33] M. Ono and B. C. Williams, "Decentralized chance-constrained finite-horizon optimal control for multi-agent systems," in *Proceedings of the 49th IEEE Conference on Decision and Control*, (Atlanta, GA), 2010.

[34] G. C. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Transactions on Automatic Control*, vol. 51, pp. 742–753, May 2006.

[35] D. P. de Farias and B. Van Roy, "On constraint sampling in the linear programming approach to approximate dynamic programming," *Mathematics of Operations Research*, vol. 29, no. 3, pp. pp. 462–478, 2004.

[36] J. Luedtke and S. Ahmed, "A sample approximation approach for optimization with probabilistic constraints," *SIAM Journal on Optimization*, vol. 19, pp. 674–699, July 2008.

[37] A. Ruszczyński, "Probabilistic programming with discrete distributions and precedence constrained knapsack polyhedra," *Mathematical Programming*, vol. 93, pp. 195–215, 2002.

[38] L. Blackmore, "A probabilistic particle control approach to optimal, robust predictive control," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, (Keystone, Colorado), August 2006.

[39] R. Platt and R. Tedrake, "Non-Gaussian belief space planning as a convex program," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Minneapolis, Minnesota), 2012.

[40] M. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.

[41] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[42] A. Bental, L. El Ghaoui, and A. Nemirovski, *Robust Optimization.* Princeton Series in Applied Mathematics, Princeton University Press, October 2009.

[43] W. Langson, I. Chryssochoos, S. Raković, and D. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, no. 1, pp. 125 – 133, 2004.

[44] A. Richards, *Robust constrained model predictive control.* PhD thesis, Massachusetts Institute of Technology, 2005.

[45] R. Cogill and S. Lall, "Suboptimality bounds in stochastic control: A queueing example," in *Proceedings of the American Control Conference*, (Minneapolis, Minnesota), June 2006.

[46] Y. Wang and S. Boyd, "Performance bounds for linear stochastic control," *Systems and Control Letters*, vol. 58, pp. 178–182, March 2008.

[47] Y. Wang and S. Boyd, "Performance bounds and suboptimal policies for linear stochastic control via LMIs," *International Journal of Robust and Nonlinear Control*, vol. 21, pp. 1710–1728, September 2011.

[48] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[49] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[50] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, pp. 846–894, June 2011.

[51] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *International Journal of Robotics Research*, vol. 8, pp. 1448–1465, December 2009.

[52] R. Alterovitz, T. Simeon, and K. Goldberg, "The stochastic motion roadmap: A sampling framework for planning with Markov motion uncertainty," in *Proceedings of Robotics: Science and Systems*, (Atlanta, GA, USA), June 2007.

[53] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, "Belief space planning assuming maximum likelihood observations," in *Proceedings of Robotics: Science and Systems*, (Zaragoza, Spain), June 2010.

[54] V. A. Huynh and N. Roy, "icLQG: Combining local and global optimization for control in information space," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Kobe, Japan), May 2009.

[55] J. Gonzalez and A. Stentz, “Using linear landmarks for path planning with uncertainty in outdoor environments,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[56] R. Pepy and A. Lambert, “Safe path planning in an uncertain-configuration space using RRT,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[57] A. Bry and N. Roy, “Rapidly-exploring random belief trees for motion planning under uncertainty,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Shanghai, China), 2011.

[58] J. van den Berg, P. Abbeel, and K. Goldberg, “LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information,” in *Proceedings of Robotics: Science and Systems*, (Zaragoza, Spain), June 2010.

[59] S. Patil, J. van den Berg, and R. Alterovitz, “Motion planning under uncertainty in highly deformable environments,” in *Proceedings of Robotics: Science and Systems*, (Los Angeles, CA, USA), June 2011.

[60] P. Missiuro and N. Roy, “Adapting probabilistic roadmaps to handle uncertain maps,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Orlando, FL), pp. 1261–1267, May 2006.

[61] B. Burns and O. Brock, “Sampling-based motion planning with sensing uncertainty,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Roma, Italy), April 2007.

[62] C. Stachniss, G. Grisetti, and W. Burgard, “Information gain-based exploration using Rao-Blackwellized particle filters,” in *Proceedings of Robotics: Science and Systems*, (Cambridge, MA, USA), 2005.

[63] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, “Planning and acting in partially observable stochastic domains,” *Artificial Intelligence*, vol. 101, no. 1-2, pp. 99 – 134, 1998.

[64] Y. Du, D. Hsu, H. Kurniawati, W. Lee, S. Ong, and S. Png, "A POMDP approach to robot motion planning under uncertainty," in *International Conference on Automated Planning & Scheduling, Workshop on Solving Real-World POMDP Problems*, 2010.

[65] H. Kurniawati, Y. Du, D. Hsu, and W. Lee, "Motion planning under uncertainty for robotic tasks with long time horizons," *International Journal of Robotics Research*, vol. 30, no. 3, pp. 308–323, 2011.

[66] G. Theocharous and L. P. Kaelbling, "Approximate planning in POMDPs with macro-actions," in *Proceedings of Advances in Neural Information Processing Systems 16*, 2003.

[67] R. He, E. Brunskill, and N. Roy, "PUMA: Planning under uncertainty with macro-actions," in *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence*, (Atlanta, GA), 2010.

[68] M. Hauskrecht, N. Meuleau, L. P. Kaelbling, T. Dean, and C. Boutilier, "Hierarchical solution of Markov decision processes using macro-actions," in *Proceedings of Uncertainty in Artificial Intelligence*, pp. 220–229, 1998.

[69] R. Sutton, D. Precup, and S. Singh, "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, pp. 181–211, 1999.

[70] M. P. Vitus and C. J. Tomlin, "Closed-loop belief space planning for linear, Gaussian systems," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Shanghai, China), May 2011.

[71] M. P. Vitus and C. J. Tomlin, "On feedback design and risk allocation in chance constrained control," in *Proceedings of the 50th IEEE Conference on Decision and Control*, (Orlando, Florida), December 2011.

[72] M. P. Vitus and C. J. Tomlin, "Belief space planning for linear, Gaussian systems in uncertain environments," in *18th World Congress of the International Federation of Automatic Control*, (Milan, Italy), September 2011.

[73] M. P. Vitus, W. Zhang, and C. J. Tomlin, "A hierarchical method for stochastic motion planning in uncertain environments," in preparation, 2012.

[74] A. Papoulis, *Probability, random variables and stochastic processes.* New York, NY: McGraw-Hill, 1965.

[75] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[76] R. T. Rockafeller and S. P. Uryasev, "Optimization of conditional value-at-risk," *Journal of Risk*, vol. 2, pp. 21–42, 2000.

[77] S. Sarykalin, G. Serraino, and S. P. Uryasev, "Value-at-risk vs. conditional value-at-risk in risk management and optimization," in *Tutorials in Operations Research*, pp. 270–294, INFORMS, 2008.

[78] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent measures of risk," *Mathematical Finance*, vol. 9, no. 3, pp. 203–228, 1999.

[79] R. T. Rockafeller, "Coherent approaches to risk in optimization under uncertainty," in *Tutorials in Operations Research*, pp. 38–61, INFORMS, 2007.

[80] J. Dhaene, M. Goovaerts, and R. Kass, "Economic capital allocation derived from risk measures," *North American Actuarial Journal*, vol. 7, no. 2, pp. 44–59, 2003.

[81] S. Basak and A. Shapiro, "Value-at-risk-based risk management: Optimal policies and asset prices," *Review of Financial Studies*, vol. 14, no. 2, pp. 371–405, 2001.

[82] C. G. de Vries, G. Samorodnitsky, B. N. Jorgensen, S. Mandira, and J. Danielsson, "Subadditivity re-examined: the case for value-at-risk," FMG discussion papers, Financial Markets Group, 2005.

[83] P. Embrechts, A. McNeil, and D. Straumann, "Correlation and dependence in risk management: properties and pitfalls," in *Risk Management: Value at*

Risk and Beyond (M. A. H. Dempster, ed.), Cambridge: Cambridge University Press, 2001.

[84] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[85] P. R. Kumar and P. Varaiya, *Stochastic Systems: Estimation, Identification, and Adaptive control, Prentice-Hall Information and System Sciences Series.* Englewood Cliffs, NJ: Prentice Hall, 1986.

[86] M. Ono, L. Blackmore, and B. C. Williams, "Chance constrained finite horizon optimal control with nonconvex constraints," in *Proceedings of the 2010 American Control Conference*, (Baltimore, Maryland), June 2010.

[87] M. P. Vitus, S. L. Waslander, and C. J. Tomlin, "Locally optimal decomposition for autonomous obstacle avoidance with the tunnel-MILP algorithm," in *Proceedings of the 47th IEEE Conference on Decision and Control*, (Cancun, Mexico), December 2008.

[88] J. Skaf and S. Boyd, "Design of affine controllers via convex optimization," *IEEE Transactions on Automatic Control*, vol. 55, pp. 2476 –2487, November 2010.

[89] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21." `http://cvxr.com/cvx`, April 2011.

[90] O. Costa, M. Fragoso, and R. Marques, *Discrete-Time Markovian Jump Linear Systems.* Springer-Verlag, 2005.

[91] L. Blackmore, A. Bektassov, M. Ono, and B. C. Williams, "Robust, optimal predictive control of jump Markov linear systems using particles," in *Hybrid Systems: Computation and Control* (A. Bemporad, A. Bicchi, and G. Buttazzo, eds.), vol. 4416 of *Lecture Notes in Computer Science*, pp. 104–117, Berlin Heidelberg: Springer-Verlag, April 2007.

[92] M. R. Garey and D. S. Johnson, *Computers And Intractability: A guide to the Theory of NP-Completeness.* New York, NY, USA: W. H. Freeman and Co., 1979.

[93] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity.* Mineola, NY, USA: Dover Publications, Inc., 1998.

[94] A. Doucet, N. De Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice.* Statistics for Engineering and Information Science, New York, NY: Springer, 2001.

[95] A. Prékopa, *Stochastic Programming.* Dordrecht, The Netherlands: Kluwer Scientific, 1995.

[96] N. Amenta, M. Bern, and M. Kamvysselis, "A new voronoi based reconstruction algorithm," in *SIGGRAPH*, (Orlando, FL), 1998.

[97] D. M. Blei and L. P. Kaelbling, "Shortest paths in a dynamic uncertain domain," in *Proceedings of the IJCAI Workshop on Adaptive Spatial Representations of Dynamic Environments*, 1999.

[98] J. Barry, L. P. Kaelbling, and T. Lozano-Perez, "Hierarchical solution of large Markov decision processes," tech. rep., Massachusetts Institute of Technology, 2010.

[99] P. Bouffard and C. Rose, "Zero-offset MPC (δu formulation) for a quadrotor UAV," May 2011.

[100] N. Vandapel, J. Kuffner, and O. Amidi, "Planning 3D path networks in unstructured environments," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Barcelona, Spain), 2005.

[101] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Shanghai, China), May 9-13 2011.