1-28-2015

Computational Techniques for Stochastic Reachability

Kendra Anne Lesser

Kendra Lesser

Candidate

Electrical and Computer Engineering

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Meeko Oishi, Chairperson

Rafael Fierro

Majeed Hayat

Lydia Tapia

Richard Scott Erwin

Computational Techniques for Stochastic Reachability

by

Kendra Lesser

B.S., Mathematics, McGill University, 2007
M.S., Applied Mathematics, University of New Mexico, 2010

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Engineering

The University of New Mexico

Albuquerque, New Mexico

December 2014

Dedication

To Matthias. You know why.

And to my parents. For always casually expecting the best.

Acknowledgments

My path through graduate school was in no way linear, and I am extremely grateful to all of the people who helped ensure that I still made it through. Mostly, I am thankful to my advisor, Meeko Oishi, for letting me work in my own way on projects I actually wanted to pursue. I am thankful to Scott Erwin, for finding applications to my work, and reminding me that systems in the real world are not two dimensional; Lydia Tapia and Nick Malone for thinking of a really neat project that lets us show off videos; Rafael Fierro for being such a friendly and supportive figure; and finally Majeed Hayat, for also being so supportive and keeping me honest when it comes to probability theory. I am also very indebted to Cristina Pereyra, who was always helpful and encouraging even after I left the math department.

Outside of school, I am grateful to the many people who kept me sane, and reminded me there is a lot more to life. Most importantly, Matthias got me back to Albuquerque and UNM. Thanks to him I have avoided many meltdowns and computer catastrophes, and have found a balance between work and fun. My parents somehow did not give up on me, and their encouragement has kept me going, even when I was not sure I wanted to. Finally, my brother Aaron was wonderful at reminding me I should probably graduate at some point.

And lastly I must thank the funding agencies, without whose financial support I could not have been in graduate school. This research was funded in part by the Air Force Office of Scientific Research through a Summer Faculty Fellowship position and through a Space Scholars Award, the National Security Agency through the Science of Security Lablet at North Carolina State University, and the National Science Foundation. This material is based upon work supported by the National Science Foundation under Grant No. CMMI-1254990, CNS-1329878, and CMMI-1335038. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Computational Techniques for Stochastic Reachability

by

Kendra Lesser

B.S., Mathematics, McGill University, 2007

M.S., Applied Mathematics, University of New Mexico, 2010

Ph.D., Electrical Engineering, University of New Mexico, 2014

Abstract

As automated control systems grow in prevalence and complexity, there is an increasing demand for verification and controller synthesis methods to ensure these systems perform safely and to desired specifications. In addition, uncertain or stochastic behaviors are often exhibited (such as wind affecting the motion of an aircraft), making probabilistic verification desirable. Stochastic reachability analysis provides a formal means of generating the set of initial states that meets a given objective (such as safety or reachability) with a desired level of probability, known as the reachable (or safe) set, depending on the objective. However, the applicability of reachability analysis is limited in the scope and size of system it can address. First, generating stochastic reachable or viable sets is computationally intensive, and most existing methods rely on an optimal control formulation that requires solving a dynamic program, and which scales exponentially in the dimension of the state space. Second, almost no results exist for extending stochastic reachability analysis to systems with incomplete information, such that the controller does not have access to the full state of the system.

This thesis addresses both of the above limitations, and introduces novel computational methods for generating stochastic reachable sets for both perfectly and partially observable systems. We initially consider a linear system with additive Gaussian noise, and introduce two methods for computing stochastic reachable sets that do not require dynamic programming. The first method uses a particle approximation to formulate a deterministic mixed integer linear program that produces an estimate to reachability probabilities. The second method uses a convex chance-constrained optimization problem to generate an under-approximation to the reachable set. Using these methods we are able to generate stochastic reachable sets for a four-dimensional spacecraft docking example in far less time than it would take had we used a dynamic program.

We then focus on discrete time stochastic hybrid systems, which provide a flexible modeling framework for systems that exhibit mode-dependent behavior, and whose state space has both discrete and continuous components. We incorporate a stochastic observation process into the hybrid system model, and derive both theoretical and computational results for generating stochastic reachable sets subject to an observation process. The derivation of an information state allows us to recast the problem as one of perfect information, and we prove that solving a dynamic program over the information state is equivalent to solving the original problem. We then demonstrate that the dynamic program to solve the reachability problem for a partially observable stochastic hybrid system shares the same properties as for a partially observable Markov decision process (POMDP) with an additive cost function, and so we can exploit approximation strategies designed for POMDPs to solve the reachability problem. To do so, however, we first generate approximate representations of the information state and value function as either vectors or Gaussian mixtures, through a finite state approximation to the hybrid system or using a Gaussian mixture approximation to an indicator function defined over a convex region. For a system with linear dynamics and Gaussian measurement noise, we show that it exhibits special properties that do not require an approximation of the information state, which enables much more efficient computation of the reachable set. In all cases we provide convergence results and numerical examples.

Contents

Contents xii

List of Figures xiii

List of Tables xviii

List of Notation xix

List of Acronyms xxi

1 Introduction 1

1.1 Motivation 1

1.2 Reachability Analysis for Hybrid Systems 3

1.3 Stochastic Hybrid Systems 4

1.4 Computational Approaches 6

1.5 Extension to Imperfect State Information 8

1.6 Contributions and Organization 10

1.6.1 Publications 11

1.6.2 Organization 13

2 Preliminaries 17

2.1 Discrete Time Stochastic Hybrid Systems 17

2.2 Reachability Analysis 20

2.3 Markov Decision Processes 23

2.4 Stochastic Optimal Control Formulations 25

3 Stochastic Reachability Analysis with Perfect Observations 29

3.1 Introduction 29

3.2 Related Work 31

3.3 Problem Formulation 31

3.4 Calculating Reach-Avoid Probabilities 32

3.4.1 Particle Approximation 33

3.4.2 Convex Chance-Constrained Approximation 35

3.4.3 Particle Approximation Using Feedback 37

3.5 Application to Spacecraft Rendezvous and
Docking 39

3.6 Summary 48

4 Stochastic Reachability Analysis with Partial Observations 49

4.1 Introduction 49

4.2 Related Work 51

4.3 Problem Formulation 52

4.3.1 Partially Observable Stochastic Hybrid System 52

4.3.2 Reachability Analysis for a PODTSHS 54

4.4 POMDP Overview 56

4.5 Reformulation of Viability Problem 58

4.5.1 Sufficient Statistic 59

4.5.2 Equivalence to Perfect State Information Problem 62

4.5.3 Sufficient Statistic with Change of Measure 65

4.5.4 Sufficient Statistic for Reachability and Reach-Avoid Problems 68

4.5.5 Relationship to Additive Cost Formulation 70

4.6 Case Studies and Numerical Issues 71

4.6.1 Temperature Regulation 71

4.6.2 Skid-Steered Vehicle 73

4.6.3 Computational Challenges 75

4.7 Summary 76

5 Computational Methods for Reachability with Partial Observations 78

5.1 Introduction 79

5.2 Related Work 81

5.3 Problem Formulation 82

5.4 Approximation Strategies for POMDPs 84

5.5 Reformulation of Problem 5.1 86

5.5.1 Validity of POMDP Formulation 87

5.5.2 Finite State Approximation 90

5.5.3 Gaussian Mixture Approximation 99

5.6 Approximate Numerical Solution with Lower Bound 107

5.6.1 Sampling from the information space 107

5.6.2 Calculating the Alpha-functions 110

5.7 Extension to Reachability and Reach-Avoid 117

5.8 Numerical Example 118

5.9 Summary 123

6 Linear Time-Invariant Systems with Noisy Observations 125

6.1 Introduction 125

6.2 Problem Formulation 128

6.2.1 System Model 128

6.2.2 Dynamic Programming Solution 129

6.3 PBVI for Verification of an LTI System 130

6.3.1 Information State 130

6.3.2 Observations 133

6.3.3 Alpha-vectors 137

6.3.4 Approximate Solution to Problem 1 138

6.4 Numerical Examples 139

6.4.1 Temperature Regulation 139

6.4.2 Anesthesia Delivery 143

6.5 Summary 147

7 Conclusion 148

7.1 Summary of Contributions 149

7.2 Future Directions 151

A Proofs of Theorems from Chapter 4 154

A.1 Proof of Theorem 4.1 154

A.2 Proof of Theorem 4.2 157

B Gaussian Mixture Results from Chapter 5 159

B.1 Information State as a Gaussian Mixture 160

B.2 Alpha-Function as a Gaussian Mixture 162

References 167

List of Figures

1.1 Artistic rendering of DART and target satellite MUBLCOM. Source: NASA [NAS07] 2

1.2 Summary of existing methods for solving stochastic reachability problems and our proposed methods. We highlight defining characteristics of each method to demonstrate how our proposed methods are distinct from existing methods and each other. 16

2.1 Example of viable, reachable, and reach-avoid sets. 21

3.1 The sets K and T in the x and y dimensions. The black lines represent the LoS cone the deputy is trying to remain within, and the green box shows the target that is close to the chief (placed at the origin). 40

3.2 Reach-avoid probability starting at various x_0 and y_0, for positive initial velocities $\dot{x}_0 = 0.01$ km/s, $\dot{y}_0 = 0.01$ km/s, using the particle approximation method with 800 particles. The dotted black line indicates the boundary of the LoS cone stemming from the origin. The area of nonzero probability is contained in the region where x_0 and y_0 are both negative, because both initial velocities are positive. 41

3.3 The sets $\mathrm{RA}^5(0.3; K, T)$ and $\mathrm{RA}^5_{0.2}(K, T)$ (i.e. the sets of all X_0 such that $\tilde{p}^5_{\mathrm{ra}}(\mathbf{x}_0; K, T) \geq 0.7$ and $\tilde{p}^5_{\mathrm{ra}}(\mathbf{x}_0; K, T) \geq 0.8$) with $\dot{x}_0 = 0.01$ km/s, $\dot{y}_0 = 0.01$ km/s in (a), and $\dot{x}_0 = -0.01$ km/s, $\dot{y}_0 = 0.01$ km/s in (b), fixed. The reach-avoid sets for $\alpha = 0.3$ and $\alpha = 0.2$ are shown for the convex approximation (black, blue), and for the particle approximation using 800 particles (red, green) methods. The dotted black line indicates the boundary of the LoS cone. RA sets are not symmetric when $\dot{x}_0$ switches from positive to negative because of the asymmetric nature of the dynamics. 42

3.4 The sets $\mathrm{RA}^5(0.3; K, T)$ and $\mathrm{RA}^5(0.2; K, T)$ (i.e. the sets of all $\mathbf{x}_0$ such that $\tilde{p}^5_{\mathrm{ra}}(\mathbf{x}_0; K, T) \geq 0.7$ and $\tilde{p}^5_{\mathrm{ra}}(\mathbf{x}_0; K, T) \geq 0.8$) with $x_0 = -0.9$ km, $y_0 = -1$ km in (a), and $x_0 = 0.9$ km, $y_0 = -1$ km in (b), fixed. The reach-avoid sets for $\alpha = 0.3$ and $\alpha = 0.2$ are shown for the convex approximation (black, blue), and for the particle approximation using 800 particles (red, green) methods. In both figures, the RA sets span only a few m/s in each direction, demonstrating the importance of accuracy in the initial velocities for rendezvous to be successful. . . . 42

3.5 Three-dimensional reach-avoid set $\mathrm{RA}^5(0.2; K, T)$, with $\dot{y}_0$ fixed at 10 m/s, generated using the particle approximation method. As x_0 ranges from negative to positive, $\dot{x}_0$ ranges from positive to negative. The cross-sections for fixed $\dot{x}_0$ shrink as x_0 becomes positive due to the asymmetric nature of the zero-input dynamics. The difference in cross-sections for x_0 positive versus negative are highlighted in Fig. 3.3. 44

3.6 The sets $RA^5_{0.1}(K,T)$ (black) and $RA^5_0(K,T)$ (red) (i.e. the sets of all X_0 such that $p^5_{X_0}(K,T) \geq 0.9$ and $p^5_{X_0}(K,T) = 1$) with $\dot{x}_0 = 0.01$ km/s, $\dot{y}_0 = 0.01$ km/s in (a), and $x_0 = -0.9$ km, $y_0 = -1$ km in (b), when using state feedback, calculated using the particle method with 500 particles. The dotted black line in (a) shows the LoS cone. The RA sets are much larger, and attain greater probabilities, than in the case of open loop control, spanning more than half of the LoS cone in (a) and a much larger range of velocities in (b). 47

5.1 Comparison of viability probabilities over varying initial distribution $\rho = \phi(x;\mu_0,1)$ and $q_0 = 0$, using the finite state approximation (a) and Gaussian mixture approximation (b). In both (a) and (b) $\delta^y = 0.5$. Fig. (a) compares probabilities for $\delta^x = 0.1$ (black dashed line) and $\delta^x = 0.01$ (red solid line). Fig. (b) compares probabilities for $I_q = 10$ (black dashed line) and $I_q = 30$ (red solid line). The refinement of δ^x and increase in I_q has a small impact on the viability probabilities. The finite state approximation estimates higher probabilities for μ_0 in the interior of K than the Gaussian mixture approximation. . . . 119

5.2 Comparison of optimal control inputs as a function of $\rho = \phi(x;\mu_0,1)$ with $q_0 = 0$, using the finite state approximation (a) and Gaussian mixture approximation (b). In both (a) and (b), $\delta^y = 0.5$. Fig. (a) compares control inputs for $\delta^x = 0.1$ (black dashed line) and $\delta^x = 0.01$ (red solid line). Fig. (b) compares control inputs for $I_q = 10$ (black dashed line) and $I_q = 30$ (red solid line), which in this case are the same. All approaches produce a thresh-hold policy that turns the heater off for $\mu_0 > 18.7$, except the finite approximation with $\delta^x = 0.1$, which turns the heater off for $\mu_0 > 18.8$. 119

5.3 Comparison between $1_K(x)$ (in black, dashed line) to RBF approximation (red, solid line) for (a) $I_q = 10$ components, (b) $I_q = 30$ components, (c) $I_q = 100$ components, and (d) $I_q = 400$ components. As the number of components increases, the approximation improves, although oscillations at the endpoints remain. 122

5.4 The L^1 error for RBF approximations to indicator function 1_K with a varying number of components I_q. As the number of components increases, the error converges towards zero. 122

6.1 Example of how truncated region I_n for the information state is updated at each time step, for an LTI system in $\mathbb{R}^2$. At time step n, I_n is intersected with K, and the resulting set is propagated through the dynamics, resulting in $I_{n+1} = \{A\overline{x} + Bu : \overline{x} \in K \cap I_n\}$. 132

6.2 The function $\|\frac{\partial \phi}{\partial y}\|$ (in black), expressed for clarity as a function of $\|f_i(x)\|$ rather than x, assuming $\mathcal{W} = [1\,,\,0; 0\,,\,1]$ and $n = 2$. The function $h(x)$ (in red, dashed) shown here as a function of $f_i(x)$, is derived from $\|\frac{\partial \phi}{\partial y}\|$ and represents a tighter Lipschitz bound on the Gaussian observation. 135

6.3 (a) The estimated probability of staying within safe set K over $N = 10$ time steps as a function of x_1, x_2 with $\sigma_0 = \mathcal{N}(\mu_0, \Sigma_0)$ and $\mu_0 = [x_1, x_2]$. (b) The optimal control input $u(0)$ as a function of μ_0. The optimal control $u(0) = 2$ is in red (i.e. heat room 2), $u(0) = 1$ in green, and $u(0) = 0$ in blue. As expected, the probability of staying within K drops towards the boundaries of K, and the optimal control is a threshhold policy that switches when the mean μ_0 falls inside certain regions. This result is computed using the *AdaptiveGridding* algorithm with $TOL = 0.05$ (computation time given in Table 6.1). 142

6.4 Probabilistic viable sets for initial distribution $\rho(x) = \phi(x; \mu_0, \Sigma_0, K)$ with $\mu_0 = [x_1, x_2, x_3]$. Each figure shows viability probabilities for μ_0 varying in x_1, x_2 and x_3 fixed at a) $x_3 = 0.0$, b) $x_3 = 1.0$, and c) $x_3 = 2.0$. The peaks at the boundaries of K are a result of initializing ρ as a truncated Gaussian limited to lying within K. This essentially provides additional information regarding the starting position of x_0 when μ_0 is near the boundary. 145

6.5 Optimal control input $u(0)$ (initial dosage of anesthesia) for initial distribution $\rho(x) = \phi(x; \mu_0, \Sigma_0, K)$ with $\mu_0 = [x_1, x_2, x_3]$ for a) $x_3 = 0.0$, b) $x_3 = 1.0$, and c) $x_3 = 2.0$. 146

List of Tables

5.1 Computation times using PBVI with finite state approximation, for varying continuous state spacing δ^x and discretized observation spacing δ^y. 121

5.2 Computation times using PBVI with Gaussian mixture approximation, for varying number of components I_q for RBF approximation to 1_K and discretized observation spacing δ^y. 122

6.1 Comparison of computation times. Adaptive gridding of $\mathcal{Y}$ is clearly more effective than a uniform grid spacing. Likewise, computational performance is degraded with discretization of σ. 141

6.2 Model Parameters from Paedfusor Data Set 143

List of Notation

$\mathbb{P}$	Probability
$\mathbb{P}^{\pi}$	Probability evaluated with respect to probability measure π
$\mathbb{E}$	Expected value
$\mathbb{E}^{\pi}$	Expected value evaluated with respect to probability measure π
$\mathbb{R}^n$	The set of all real-valued numbers in n-dimensional space
$\emptyset$	The null (empty) set
$\exists$	There exists
$\forall$	For all
$\wedge$	Logical "and" operator
$\vee$	Logical "or" operator
$\implies$	Implies
$\cap$	Intersection set operator
$\cup$	Union set operator
$\subseteq$	Subset set operator
A^c	Complement of set A
$A\backslash B$	Set A excluding all elements also contained in set B
$A \times B$	Cartesian product of sets A and B
A^n	Cartesian product of set A with itself n times, $A \times A \times \ldots \times A$
$\|A\|$	Cardinality of set A
$\|x\|$	Absolute value of $x \in \mathbb{R}^n$
$\mathcal{L}(A)$	Lebesgue measure of set A
$\mathcal{B}(\mathcal{S})$	The Borel σ-algebra on space $\mathcal{S}$
$\\|\cdot\\|_m$	m-norm of either a vector or function

L^p	The space of functions whose p-norm is finite
$\langle f, g\rangle$	Inner product of f and g (either functions or vectors)
$\mathcal{N}(\mu, \Sigma)$	Gaussian (normal) distribution with mean μ and covariance Σ
$\phi(x; \mu, \Sigma)$	Gaussian pdf evaluated at x, with mean μ and covariance Σ
$\phi(x; \mu, \Sigma, I)$	Truncated Gaussian pdf evaluated at x, with mean μ, covariance Σ, and support I
$\Phi(x)$	Gaussian cdf evaluated at x (with mean 0 and identity covariance)
$1_A(x)$	The indicator function for set A
A^T	Transpose of matrix A
A^{-1}	Inverse of matrix A
A^{-T}	Inverse of the transpose of matrix A

List of Acronyms

cdf	Cumulative distribution function
CWH	Clohessy-Wiltshire-Hill
DTSHS	Discrete time stochastic hybrid system
iid	Independent and identically distributed
LoS	Line-of-sight
LTI	Linear time-invariant
MDP	Markov decision process
MILP	Mixed integer linear program
PBVI	Point-based value iteration
pdf	probability density function
PODTSHS	Partially observable discrete time stochastic hybrid system
POMDP	Partially observable Markov decision process
RBF	Radial basis function

Chapter 1

Introduction

1.1 Motivation

The scale and complexity of modern control systems makes the analysis and design of controllers equally complex. In particular, the push towards fully automated systems, ranging from vehicles to medical devices to traffic flow and regulation, places an increasingly high demand on the performance of the controllers designed for automation. Many of these systems are classified as "safety-critical," meaning that the cost of failure is deemed unacceptable, whether that cost is in property, in the loss of human life, or both.

The ability to assess the performance of such systems, and to determine whether they meet all desired specifications established in their design, is therefore paramount. Indeed, controllers should be designed to meet all specifications, and in particular to meet safety requirements when labeled as safety-critical.

For example, there is increasing interest in the development of unmanned spacecraft that can perform automated maneuvers, particularly in coordination with other spacecraft such as rendezvous, docking, and holding formation patterns. In 2005 NASA launched DART, a spacecraft designed to demonstrate automated docking and close proximity maneuvers between another spacecraft [NAS07]. The mission ended

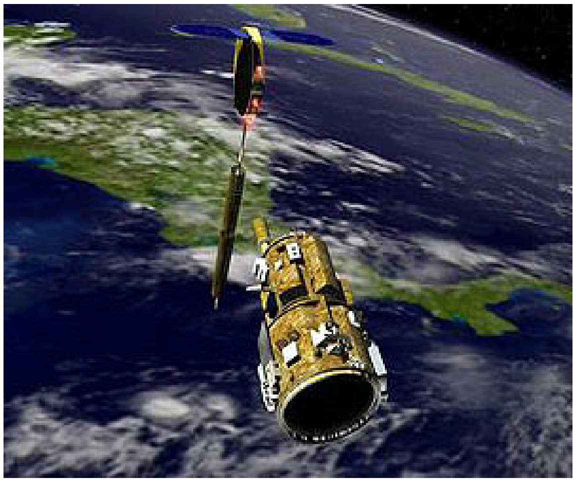


Figure 1.1: Artistic rendering of DART and target satellite MUBLCOM. Source: NASA [NAS07]

prematurely after DART collided with the other craft, costing millions of dollars. One of the reasons cited for the failure was an inadequate guidance, navigation, and control software development process. One recommendation was that "simulations and math models used to validate flight software must be verified and validated to the same rigorous level as the flight software itself" [NAS06].

Although it was a software failure caused by human error, it does highlight the need for mathematical models of control systems that are able to adequately verify the performance of the system in practice, and the need in general for sophisticated verification tools to ensure that such costly failures do not occur.

Another example of a safety-critical system is the automated, closed-loop control of anesthesia delivery [DMA09]. The anesthetic must be delivered in such a way as to stay within pre-specified therapeutic bounds on concentration levels subject to limitations on how quickly the drug can be delivered (infusion rate). Further complications arise from the uncertain nature of individual patient response, and from monitoring drug levels and depth of anesthesia [BMSS06]. In order for automated anesthesia delivery to be accepted as a safe practice, thus meeting regulatory standards, theoretical guarantees of its ability to satisfy safety specifications are essential.

We are concerned in this thesis with designing controllers and analyzing performance, particularly in relation to safety specifications, of certain classes of systems, such as space docking maneuvers and automated anesthesia delivery. In these two examples, the system benefits from safety verification, in addition to controller synthesis that assures safety specifications are met. However, there are a wide range of examples for which the techniques presented in this thesis are applicable. Analysis and control for safety specifications can and has been applied to areas from motion planning algorithms to routing packets through wireless networks. The stochastic techniques we focus on are also desirable in general for cyber-physical systems, whose integration of modal logic into physical processes makes them both complex to model with certainty and often safety-critical. Generating probabilistic assurances of safety that are not overly conservative and that can adequately capture behavior and performance characteristics are essential for such systems, and is the focus of this dissertation.

1.2 Reachability Analysis for Hybrid Systems

We consider safety requirements in the form of bounds on the state of the system, so that if the state falls outside of some predefined region of the state space, it is considered unsafe. Our approach employs reachability analysis, which is commonly used to determine whether a control system satisfies given specifications, such as safety.

There are three properties that fall under the general concept of reachability analysis, and all are intricately related. First is *reachability*, which refers to controlling the state of the system to a given target region within some time horizon. The second is *viability*, alternately referred to as *safety*, which refers to controlling the state of the system to remain within predefined safe parameters, or a set of states that has been deemed safe. The third is *reach-avoid*, which combines reachability and viability objectives to control the state of the system to reach a target region while in the interim remaining within a safe region. As we will describe in subsequent chapters, these objectives can all be studied within the same framework.

In particular, reachability analysis is concerned with generating the set of *all* initial states that satisfy the given objective (i.e. the set of all initial states that will reach the target set, or the set of all initial states that will remain within the safe region) referred to as the reachable set, viable set, or reach-avoid set, depending on the objective of interest. For control systems, reachability analysis takes possible control inputs into account by determining whether a controller exists to satisfy the objective. A related problem of *invariance* assesses whether the state of the system remains within a given region *for all* possible controllers. Dependent on the method used to compute reachable sets, reachability analysis can also be used to synthesize controllers to meet the given specifications. Synthesizing a controller to meet the reachability specifications proves its existence.

Reachability analysis is often addressed in the context of hybrid systems (e.g [LTS99], [GLQ06]). A hybrid system provides a general modeling framework well-suited for a wide range of applications. It allows for versatile dynamics that incorporate codependent discrete and continuous states, often exhibited in systems that may switch between different modes of operation. Safety verification and controller synthesis for systems that integrate hierarchical mode logic with nontrivial physical dynamics (often modeled by differential equations) is both highly complex and essential, given that many modern control systems exhibit such behavior. There is a large body of literature on hybrid system modeling and control (e.g. [ASL93], [BBM98], [AHS96]), and hybrid systems have been used as a modeling framework for applications ranging from air traffic management [TPS98] to analyzing motor skills of patients with Parkinson's disease [OMAM11].

1.3 Stochastic Hybrid Systems

Most systems are subject to some form of uncertainty, whether from wind patterns affecting aircraft flight, the sometimes unpredictable response of a patient to anesthesia, or from nonlinearities and other factors that are overly complex, or simply not known, and therefore cannot be included in our models. Especially when verifying safety and other specifications of the system, these uncertainties should be taken into account in order to have accurate assessments of system performance (or to recognize when we *cannot* have accurate assessments). Modeling a system as stochastic allows the state to evolve randomly, but with some notion of which states are mostly likely to occur. This is distinct from a model that incorporates uncertainty in the form of a disturbance whose behavior is completely unknown, but whose impact on the model may be confined to a certain range of possibilities.

Reachability analysis for uncertain hybrid systems typically treats the disturbance as a bounded input to the model. The disturbance acts as an adversary, behaving in direct opposition to the controller and the given objective. This enables a dynamic game formulation of the reachability problem [KV02], [GLQ06], [DKS$^+$13], and the controller and reachable set are then robust to the worst-case manifestation of the uncertainty.

A stochastic hybrid system considers stochastic transitions in either the discrete mode, the continuous dynamics, or both. Rather than assume worst-case scenarios, we can assign likelihoods to different possible hybrid states. One advantage of a stochastic hybrid system (SHS) model is in the equivalent notion of probabilistic reachability. Rather than generate the set of initial states that reach or remain within a given region of the state space, we can produce the set of initial states that do so with a certain probability. This is particularly useful when uncertain events may drive the system into an unsafe region, but with such a small likelihood that they can be ignored. Depending on the application, a 99 percent probability of satisfying the reachability objective is most likely tolerable. We therefore choose to confine our attention to stochastic systems.

One of the first SHS models was developed in [HLS00], and describes the continuous state process with a stochastic differential equation. Mode transitions remain deterministic, but occur at *stopping times* associated with the diffusion process (see [Res02] for a description of stopping times). The stopping times, and stochastic transitions occurring at each stopping time, lead to the notion of an *embedded Markov process*, over which some properties of the original SHS can be verified. This model was extended in [PH06] to include stochastic mode switching as well. A general SHS model was proposed in [BL06] using a new concept they term a Markov string, which is essentially a sequence of Markov processes.

The additional complications from introducing stochastic dynamics into the hybrid system framework are significant. Combining a jump process (for mode switching) and a diffusion process (for continuous state evolution) leads to measurability concerns regarding events of interest, such as identifying stopping times and reachability events. These concerns are addressed in [Buj04], which uses Dirichlet forms associated with a right-Markov process to study reachability, and again in [BL07], which employs martingale theory and characterizes reachability in terms of variational inequalities. Although these papers establish some theoretical foundations for posing reachability as a well-defined problem over an SHS with "nice" properties, their use for practical purposes is at present limited.

These concerns led to the development of an equivalent *discrete time* stochatic hybrid system (DTSHS) [APLS08]. In discrete time, many of the measurability complications arising from a continuous time model disappear, and numerical solutions to reachability problems can be obtained. For this reason, we are concerned only with discrete time systems in this thesis, although as will be seen, computational methods remain limited in the size and scale of problems they can address.

1.4 Computational Approaches

The reachability problem can be equivalently posed as an optimal control problem. This is true for reachability in general, and is not specific to hybrid systems, deterministic systems, etc. The optimal control formulation introduces a cost function that assigns a reward of one to states that remain safe, or reach the target set, and assigns a reward of zero to all other states (see, e.g. [LTS99], [TLS00]). Alternately, the cost function can be formulated as a signed distance function to the target set (for backwards reachability), so that all states for which the cost function is negative are inside the reachable set [TMBO03]. Through the optimal control formulation, the vast literature on computing optimal costs and optimal controllers can be utilized.

Specifically, the reachable set is characterized as the zero level-set of the value function associated with the optimal control objective [TMBO03], [Lyg04], [MBT05]. The value function is the viscosity solution to a Hamilton-Jacobi equation (a first order partial differential equation), which arises from Bellman's principle and the dynamic programming formulation utilized in optimal control problems [Ste94]. Computational methods for generating the level-sets are presented in [MT02], [Mit08]. All of the above results consider continuous time systems.

The Hamilton-Jacobi formulation can in some cases still be applied to reachability analysis for continuous time stochastic systems, as in [MT05] and [KR06], although neither allows for a control input. Switching to discrete time, however, makes the inclusion of a control input easier. The reachability problem for a controlled discrete time stochastic system (hybrid or otherwise) can again be expressed as an optimal control problem, now drawing upon the related literature for Markov decision processes and discrete time stochastic optimal control [Ber05], [BS96]. Because of the stochastic nature of the state, the cost function is modified from the binary cost function described above, so that the reward of zero or one is placed inside an expected value, which in turn gives the probability of safety or reachability, starting from some initial state. Dynamic programming formulations for the safety/reachability problem, reach-avoid problem, and a dynamic game formulation, are given in [APLS08], [SL10], and [DKS$^+$13], respectively.

One significant drawback to the dynamic programming formulation (for both stochastic and deterministic systems), is that its numerical implementation requires a discretization of the state space. The computation time scales exponentially in the dimension of the state space, a problem known as the "curse of dimensionality." Hence for systems lying in higher dimensions, i.e. for hybrid systems with a continuous state that lies in $\mathbb{R}^n$ for n greater than three or four, computation time to produce the reachable or viable set is so great that a solution is essentially unattainable. Alternative methods for generating reachable and viable sets are therefore desirable.

For deterministic systems some methods rely on over- or under-approximating the reachable set using shapes such as ellipses [KV07] or zonotopes [Gir05], or characterizing the convex reachable set using support vectors [GG10], although all of these methods require linear system dynamics. More recently, these results were extended to generating viable sets as well [MKM$^+$13], [KO].

Alternately, there has been a great deal of work on producing abstractions for both deterministic and stochastic systems, which are simpler to analyze but exhibit the same properties as the original system (for instance [Gir12], [ADB11]). Abstractions for reachability analysis of stochastic hybrid systems are often a finite state approximation to the original system, and dynamic programming is then used for the discretized system [AKLP10], [SA13]. Finite state abstractions can also be used as inputs to model-checking tools such as PRISM [KNP11], that have been developed to verify properties of finite state systems, such as Markov decision processes, using dynamic programming and abstraction refinement [KKNP10]. These model-checking techniques are still limited by the size of the finite state abstraction, and are not amenable to complex systems that require a large number of finite states to generate accurate reachability probabilities.

Finally, approximate dynamic programming techniques that employ Monte-Carlo sampling and basis function approximations to the value function generated by the dynamic program are possible, but generally lack convergence results or bounds on the relation between the approximation and the true reachable set [KSS$^+$13].

1.5 Extension to Imperfect State Information

A significant portion of this thesis is devoted to computational considerations of discrete time stochastic hybrid systems with the added constraint that the controller does not have access to the true state of the system. Most of the computational results for reachability analysis of even deterministic hybrid systems are limited to low dimensional problems, and other approaches that try to mitigate the effects of the curse of dimensionality are restricted to very specific classes of systems (i.e. linear). Adding an observation process further complicates an already difficult problem.

We specifically consider a stochastic observation process. This is particularly useful for systems that rely on sensors to take state measurements that may be inaccurate or corrupted by noise. Spacecraft rendezvous and docking procedures are a prime example of a system that must be carefully controlled in the presence of limited, noisy sensor readings.

When the state is not perfectly observed, some form of estimate must be generated that allows for optimal control of the system subject to the limited information available. Unlike linear quadratic Gaussian (LQG) regulators, in which an estimate of the state can be used to produce an optimal control law, state estimation cannot in general be considered separately from the control problem. There has been extensive work on hybrid state estimation without considering the optimal control problem simultaneously, i.e. where the main objective is to reconstruct the hybrid state as accurately as possible given a noisy or incomplete measurement (see for instance [HW04], [LH12], [KKZ03]).

Almost no work has been done, however, on reachability analysis for a controller with access only to an observation process (an optimal control problem). The work that has been done mainly focuses on deterministic hybrid systems subject to unknown (but not stochastic) disturbances. A hybrid system with hidden modes (unknown to the controller) is considered in [VdV12], and treats estimation of the mode separately to generate a non-deterministic information state that includes the mode estimate, to determine the set of initial states that cannot be controlled to stay within the safe region. A more general uncertain system with a form of input/output order preservation between states and observations (the output must be at most two-dimensional and bounded by extremal trajectories) is considered in [GdV14], where an information state is generated that represents the set of all possible states the system could be in given the observed output, and the output is controlled to remain outside of an unsafe region.

For a stochastic observation process, we cannot in general produce a finite set of possible states that could arise given the observation. Instead, we want to consider the likelihood of possible states given the observation. Stochastic optimal control of a partially observable system uses a sufficient statistic to condense all necessary information for control of the system, producing the information state [Ber05]. A dynamic program over the information state can then be formulated, as if we were solving an optimal control problem for a system with perfect information. The technique of producing an information state, and solving an equivalent problem over the information state, however, is similar for both the uncertain systems in [VdV12], [GdV14], and for a stochastic system.

Two problems arise when solving the reachability problem for a stochastic system with partial observations. The first is that the information state is a probability distribution. Dynamic programming requires iteration over all possible states of the system to generate the value function evaluated at all of those states; iterating over all possible probability distributions is infeasible. The second problem is that the reachability cost function is multiplicative in nature, meaning that the cost of being in past states affects the cost function at the current time step. For an additive cost function, the information state is the probability distribution of the current state of the system conditioned on all past observations and control inputs [Ber05]. The cost function for the reachability problem is non-additive, and so the conditional distribution of the state is no longer sufficient for optimal control. These concerns are the focus of a significant portion of this thesis, and are described in greater detail in subsequent chapters.

1.6 Contributions and Organization

The main focus of this thesis is on the computation of reachable and viable sets for stochastic systems, especially in the presence of partial observations. We make several important contributions to the advancement of reachability analysis as a verification tool for more complex and realistic systems, i.e. systems of higher dimension and systems that rely on sensor measurements and observations for control. These contributions are described in detail in the following chapters, and summarized here.

1. Development of two novel optimal control formulations to compute stochastic reachable sets for moderate dimensional systems with linear time-invariant (LTI) dynamics: 1) Particle approximations that produce a mixed integer-linear program, and 2) Convex chance-constrained optimization. We apply both techniques to a four-dimensional spacecraft rendezvous problem.
2. Derivation of an information state and dynamic programming formulation to generate stochastic reachable sets and controllers for a partially observable discrete time stochastic hybrid system.
3. Development of two novel computational techniques for generating approximate stochastic reachable sets for partially observable discrete time stochastic hybrid systems: 1) Finite state approximations, and 2) Gaussian mixture approximations. Both methods are proven to provide convergence to the original reachable set, and their performance is demonstrated on a temperature regulation problem.
4. Derivation of the information state as equal to a truncated Gaussian distribution for an LTI system with Gaussian measurement noise. An adaptive gridding scheme is also established for the observations. Computational results are provided for a two-dimensional temperature regulation problem and a three-dimensional anesthesia delivery problem.

Figure 1.2 gives an overview of both existing computational methods for stochastic reachability analysis and the methods proposed in this dissertation, and highlights distinguishing features of each method.

1.6.1 Publications

Most of the work presented here has previously been published, or submitted for publication. The work of Chapter 3 is presented in:

[i] K. Lesser, M. Oishi, and R.S. Erwin. Stochastic reachability for control of spacecraft relative motion. In *IEEE Conference on Decision and Control*, pages 4705 - 4712, 2013.

The results of Chapter 4 are published in

[ii] K. Lesser and M. Oishi. Reachability for partially observable discrete time stochastic hybrid systems. *Automatica*, 50(8):1989-1998, 2014.

The results of Chapter 5 have been submitted for publication in *IEEE Transactions on Automatic Control* and for presentation at *Hybrid Systems: Computation and Control* as part of *CPS Week*.

[iii] K. Lesser and M. Oishi. Approximate verification of partially observable discrete time stochastic hybrid systems. *IEEE Transactions on Automatic Control*, 2014. Submitted, under review.

[iv] K. Lesser and M. Oishi. Finite state approximation for verification of partially observable stochastic hybrid systems. Submitted to *Hybrid Systems: Computation and Control*, 2015.

The results of Chapter 6 have been submitted for presentation at the *American Control Conference*.

[v] K. Lesser and M. Oishi. Computing probabilistic viable sets for partially observable systems using truncated Gaussians and adaptive gridding. Submitted to *American Control Conference*, 2015.

Although not discussed in this dissertation, we have applied stochastic reachability techniques to several relevant and well-motivated problems. The first is the problem of assessing routing performance and generating packet delivery guarantees in multi-hop wireless network routing. By modeling a wireless network as a dynamical system whose state includes the availability of links in the network, we were able to apply stochastic reachability analysis to the system in order to generate probabilistic guarantees of packet delivery, as well as an optimal packet routing scheme.

[vi] T. Biswas, K. Lesser, M. Oishi, and R. Dutta. Using linear system reliability to obtain theoretical understanding of wireless routing. To appear in *The IEEE Global Communications Conference*, 2014.

The second is a motion planning problem in the presence of stochastically moving obstacles. We were able to generate stochastic reachable sets that indicate the pairwise probability of collision between a controlled robot and a stochastically moving obstacle, and use these pairwise probabilities to either weight edges of a roadmap (where a higher weight indicates a higher probability of collision) or to generate potential fields that steer the robot away from areas of high probabilities of collision. The robot can then navigate through an environment with hundreds of obstacles with a high probability of success.

[vii] H.T. Chiang, N. Malone, K. Lesser, M. Oishi, and L. Tapia. Aggressive moving obstacle avoidance using a stochastic reachable set based potential field. In *The International Workshop on Algorithmic Foundations in Robotics*, 2014.

[viii] N. Malone, K. Lesser, M. Oishi, and L. Tapia. Stochastic reachability based motion planning for multiple moving obstacle avoidance. In *Hybrid Systems: Computation and Control*, pages 51 - 60, 2014.

1.6.2 Organization

Chapter 2 provides an overview of some of the main themes of the dissertation, including a more detailed description of stochastic reachability analysis, and how it is formulated and solved as an optimal control problem. We define a discrete time stochastic hybrid system, and also review Markov decision processes (MDPs), since the optimal control of a discrete time SHS is directly related to the optimal control of an MDP.

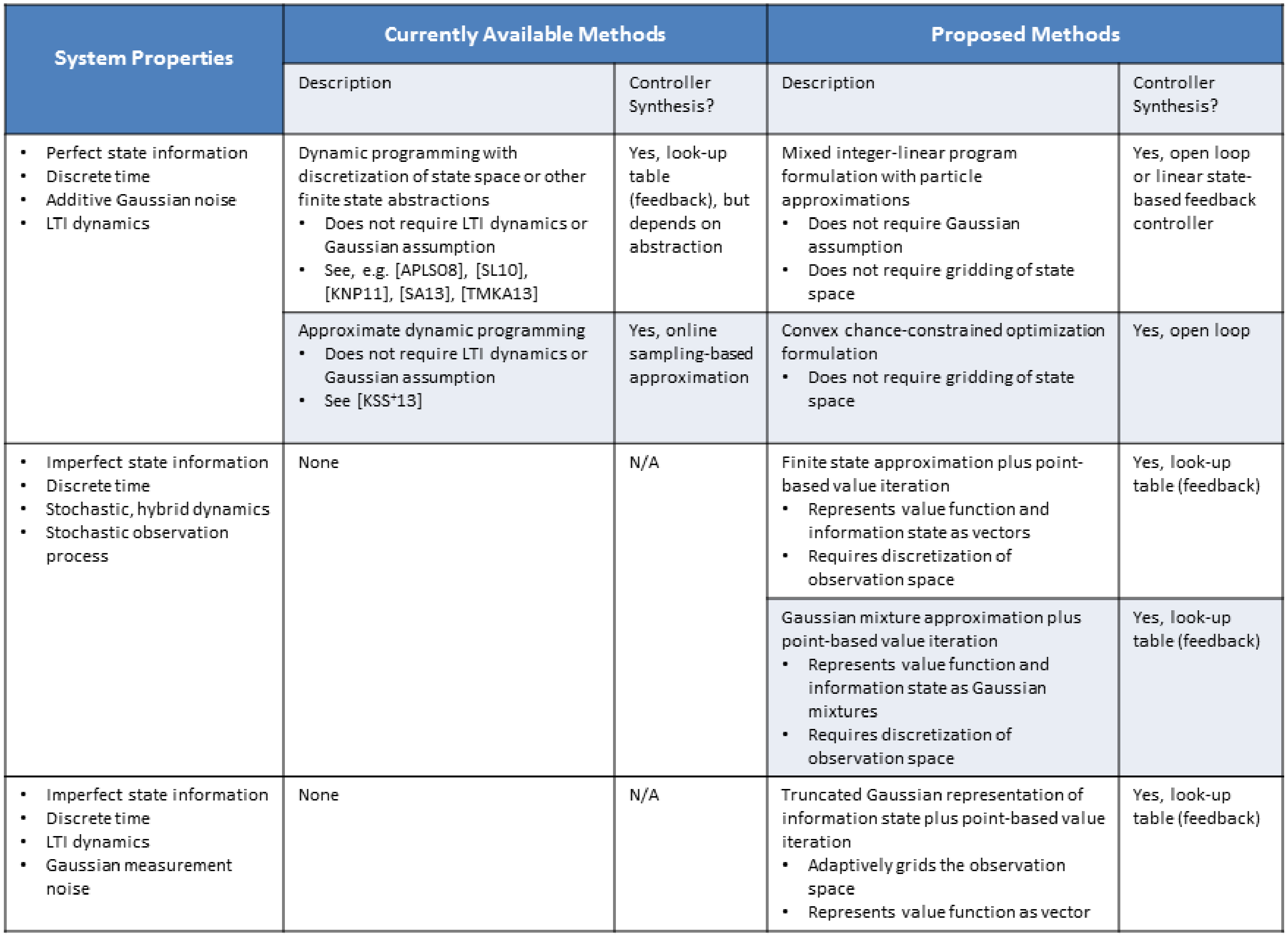
In Chapter 3 we consider a stochastic system with linear dynamics and perfect state information. We present two computational alternatives to dynamic programming for reachability analysis, which enable reachable set calculation for systems whose state is greater than three dimensional. The methods considered are a) particle approximations to reachability probabilities (i.e. a sampling based method) and b) conversion of the reachability probability to a chance constraint inside of a stochastic optimization problem. The work of Chapter 3 is motivated by a spacecraft rendezvous and docking application, and numerical results are given for a simple docking example.

We switch focus in Chapter 4 to reachability analysis subject to a stochastic observation process. We give additional background information on partially observable stochastic hybrid systems and their optimal control, as well as an introduction to partially observable Markov decision processes (POMDPs). As with MDPs, the optimal control of a POMDP is directly related to the optimal control of a partially observable discrete time stochastic hybrid system. The theoretical foundations for formulating the reachability problem under partial observations are then presented, by deriving an information state and dynamic programming formulation over the information state. We prove equivalence between the dynamic program over the information state and the original reachability problem.

The information state and dynamic program are not directly useful for computation of reachable sets. We therefore focus in Chapter 5 on how the formulation of Chapter 4 can be used to generate approximate reachable and viable sets. We consider two approximation strategies. The first generates a finite state approximation to the SHS, similarly to some of the abstraction methods mentioned above. The second introduces a Gaussian mixture approximation for both the information state and value function. Both methods then utilize an existing approximation strategy for POMDPs based on sampling information states.

Chapter 6 provides improved numerical results as compared to the previous chapter by considering a subclass of systems with linear dynamics and no process noise (only the observations are stochastic). Considering this type of system allows us to compute the reachable set more efficiently, by representing the information state as a truncated Gaussian and employing an adaptive discretization scheme for the observations. The adaptive grid is designed to reduce the number of finite observations we consider, while simultaneously minimizing the error in the approximation. We are able to implement the new approach on an example of automated anesthesia delivery.

Finally, concluding remarks and a discussion of future directions are given in Chapter 7. Supplementary material to Chapter 4 is given in Appendix A, which contains the proof of Theorem 4.1 and Theorem 4.2. Appendix B contains supplementary material for Chapter 5, including the proof of Lemma 5.6 and Lemma 5.7, as well as a complete description of how Gaussian mixture approximations are generated for the information state and value function.



System Properties	Currently Available Methods		Proposed Methods	
	Description	Controller Synthesis?	Description	Controller Synthesis?
• Perfect state information • Discrete time • Additive Gaussian noise • LTI dynamics	Dynamic programming with discretization of state space or other finite state abstractions • Does not require LTI dynamics or Gaussian assumption • See, e.g. [APLS08], [SL10], [KNP11], [SA13], [TMKA13]	Yes, look-up table (feedback), but depends on abstraction	Mixed integer-linear program formulation with particle approximations • Does not require Gaussian assumption • Does not require gridding of state space	Yes, open loop or linear state-based feedback controller
	Approximate dynamic programming • Does not require LTI dynamics or Gaussian assumption • See [KSS+13]	Yes, online sampling-based approximation	Convex chance-constrained optimization formulation • Does not require gridding of state space	Yes, open loop
• Imperfect state information • Discrete time • Stochastic, hybrid dynamics • Stochastic observation process	None	N/A	Finite state approximation plus point-based value iteration • Represents value function and information state as vectors • Requires discretization of observation space	Yes, look-up table (feedback)
			Gaussian mixture approximation plus point-based value iteration • Represents value function and information state as Gaussian mixtures • Requires discretization of observation space	Yes, look-up table (feedback)
• Imperfect state information • Discrete time • LTI dynamics • Gaussian measurement noise	None	N/A	Truncated Gaussian representation of information state plus point-based value iteration • Adaptively grids the observation space • Represents value function as vector	Yes, look-up table (feedback)

Figure 1.2: Summary of existing methods for solving stochastic reachability problems and our proposed methods. We highlight defining characteristics of each method to demonstrate how our proposed methods are distinct from existing methods and each other.

Chapter 2

Preliminaries

We begin with a general overview of the type of problem we wish to solve, the dynamical system model over which it is posed, and a solution framework from an optimal control perspective. We first describe discrete time stochastic hybrid systems in Section 2.1. We then discuss the reachability problem and its variants in Section 2.2, give an overview of Markov decision processes in Section 2.3, and close with how reachability can, in theory, be solved in a similar manner to Markov decision processes using optimal control techniques, in Section 2.4.

2.1 Discrete Time Stochastic Hybrid Systems

The term "stochastic" implies a system with uncertainty, and in particular with uncertainty that may be quantified in the form of a probability distribution. In other words, a stochastic system or process progresses in time randomly [Res02], but all outcomes can be assigned a likelihood of occurring based on some prior knowledge or assumptions. Where the uncertainty comes from, and how it is modeled, varies. Uncertainty may be in the model itself, or in external factors beyond our control (wind affecting an aircraft, demand for a product on any given day, etc.). Incorporating stochastic uncertainty brings a level of robustness to a model without necessarily being overly conservative, and is accompanied by a host of well-developed probabilistic tools to help in its analysis.

Stochastic hybrid systems provide a modeling framework well-suited to a wide range of applications, including complex and interconnected systems. Stochastic hybrid systems couple physical processes, that are often naturally expressed in continuous spaces, with discrete or finite state processes, often associated with computer logic and switching modes. The discrete state may affect the evolution of the continuous dynamics, and the continuous dynamics may affect when the discrete state changes. Both the discrete and continuous dynamics may be characterized by stochastic kernels, the product of which determines the stochastic transition kernel governing the combined discrete/continuous state of the system.

We present a discrete time stochastic hybrid system (DTSHS) model, adapted from [APLS08].

Definition 2.1. *(Discrete Time Stochastic Hybrid System). A DTSHS is a tuple* $\mathcal{H} = (X, \mathcal{Q}, \mathcal{U}, T_x, T_q)$ *where*

1. $\mathcal{X} \subseteq \mathbb{R}^n$ *is a set of continuous states*
2. $\mathcal{Q} = \{q_1, q_2, ...q_{N_q}\}$ *is a finite set of discrete states with cardinality* N_q*, with* $\mathcal{S} = \mathcal{X} \times \mathcal{Q}$ *the hybrid state space*
3. $\mathcal{U}$ *is a compact Borel space which contains all possible control inputs affecting discrete and continuous state transitions*
4. $T_x : \mathcal{B}(\mathbb{R}^n) \times \mathcal{Q} \times \mathcal{S} \times \mathcal{U} \to [0,1]$ *is a Borel-measurable stochastic kernel which assigns a probability measure to* x_{n+1} *given* $s_n = (x_n, q_n), u_n, q_{n+1}$ *for all n:* $T_x(dx_{n+1} \in B | q_{n+1}, s_n, u_n)$ *where* $B \in \mathcal{B}(\mathbb{R}^n)$*, the Borel* σ*-algebra on* $\mathbb{R}^n$
5. $T_q : \mathcal{Q} \times \mathcal{Q} \times \mathcal{U} \to [0,1]$ *is a discrete transition kernel assigning a probability distribution to* q_{n+1} *given* q_n, u_n *for all n:* $T_q(q_{n+1} | q_n, u_n)$

The restriction to Borel spaces and functions ensures that integration of the transition kernels T_x and T_q over the hybrid state space $\mathcal{S}$ is well defined. Kernels T_x

and T_q can be combined for ease of notation to produce the hybrid state transition kernel

$$\tau_s(ds' \mid s, u) = T_x(dx' \mid x, q, u, q')T_q(q' \mid x, q, u), \tag{2.1}$$

with $\sum_{q' \in \mathcal{Q}} \int_{\mathcal{X}} \tau_s(dx', q'|x, q, u) = 1$. The discrete state q_{n+1} update depends on q_n, x_n and u_n, and the continuous state x_{n+1} update depends on x_n, u_n, and according to the specific problem may also be governed by q_n, q_{n+1}, or both. For ease of notation, we assume that the discrete state updates first, and the updated discrete state affects the continuous state, i.e. that $T_x(dx_{n+1}|x_n, u_n, q_{n+1})$, although modifying T_x to include q_n would not alter any subsequent results.

The uncertainty in the evolution of both x and q is defined by the functions T_x and T_q, which are problem specific. These functions will be defined explicitly as necessary in subsequent chapters, dependent on the specific problem to be analyzed.

The control inputs u are chosen according to a *policy*, which maps observable outcomes to control inputs. When the state of the system is perfectly observed and available to the controller at all times, the policy maps states to control inputs. A Markov policy chooses control inputs based on information available at the current time step only, and is associated with generating optimal control inputs for Markov decision processes [Ber05], (Section 2.3).

Definition 2.2. *For a DTSHS $\mathcal{H}$, a Markov policy π for some time horizon N is a sequence of functions, $\pi = (\mu_0, \ldots, \mu_{N-1})$, such that $\mu_n : \mathcal{S} \to \mathcal{U}$. The policy is said to be* stationary *if $\mu_n = \mu$ for all $n \in [0, N]$, i.e. the mapping does not depend on the time.*

The set of all Markov policies is denoted Π. The optimal policy for an optimal control problem over a finite time horizon is generally non-stationary [Put05], and we will mostly consider only non-stationary policies. A Markov policy is equivalent to a simple feedback control law, because it is based solely on the state of the system at any time. A policy representing an open loop problem would map a single state s_0 to a vector of control inputs, $\mu(s_0) = [u_0, \ldots, u_{N-1}]$. An open loop formulation will be used in Chapter 3 but otherwise Definition 2.2 is assumed.

For a fixed policy π, the DTSHS $\mathcal{H}$ evolves as a stochastic process $\{s_n = (x_n, q_n)\}_{n=0}^{N}$. The transition kernel τ_s assigns a likelihood to state s' conditioned only on s and u, and so the DTSHS as defined is specifically a controlled Markov process. As shown in [APLS08], the process $\{s_n\}$ is a Markov process defined on space $\Omega = \mathcal{S}^N$ endowed with σ-algebra $\mathcal{B}(\Omega)$ and probability measure $\mathbb{P}^{\pi}$, which is uniquely defined dependent on transition kernel τ_s, control policy $\pi \in \Pi$ and initial state s_0.

The DTSHS $\mathcal{H}$ is equivalent to a Markov decision process [Put05] with state space $\mathcal{S}$, control space $\mathcal{U}$, and transition function τ_s. This equivalence will be used repeatedly throughout the remaining chapters, and in particular will be exploited when examining a partially observable DTSHS in Chapters 4 - 6.

2.2 Reachability Analysis

We would often like to verify certain properties of $\mathcal{H}$, i.e., if there are states deemed "unsafe", is it possible for the state to become unsafe when initialized from various starting conditions? Similarly, if there are desirable states that should be reached, are these states indeed reachable from various starting conditions? Can we characterize the set of all initial states that remain safe for a given time horizon, or will reach a target set of states in a given time horizon?

Verification of such safety and performance specifications can be framed under the guise of reachability analysis, which is a term used generally to refer to several system properties that can all be verified using the same or similar techniques. Specifically, we present three related terms: Viability, reachability, and reach-avoid.

Viability (sometimes referred to as safety verifiction) addresses the problem of determining whether the state of a system will remain within a predefined region of the state space for a given time horizon. Reachability refers to whether the state of a system can reach a predefined target region within a given time horizon. Reach-avoid is a combination of the two, and determines whether the state of the system can reach a target set while avoiding an unsafe set of states. The set of all initial states satisfying these properties are denoted the viable, reachable, and reach-avoid sets. Fig. 2.1 shows an example of all three sets.

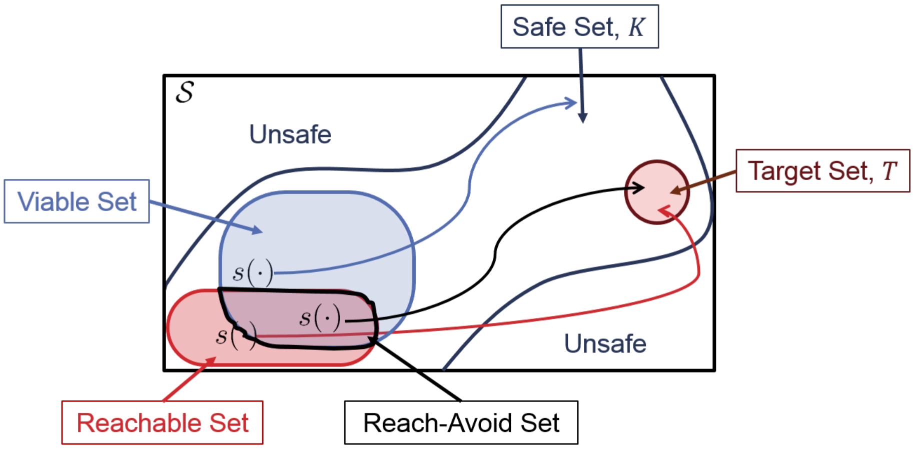


Figure 2.1: Example of viable, reachable, and reach-avoid sets.

The added advantage of reachability analysis is that it provides a means for controller synthesis as well, when the system being studied has a control input. The question can be posed as: "Does there exist a controller for which the desired specification (safety, reaching a target, etc.) can bet met?" For the stochastic setting, reachability analysis requires an additional specification, however. Rather than asking whether the specification is met (i.e. a controller exists), with a yes/no answer, the question must be posed in terms of probabilities. For some applications we would like to guarantee safety or reachability with probability equal to one. In other applications, such a requirement may be overly conservative, and a threshold probability of $1 - \alpha$ may be tolerated. In this sense, probabilistic reachability is flexible, and can be more informative than in the deterministic case if probabilities of failure are low but nonzero.

Since we want to determine whether a controller exists such that the controlled system satisfies some property with a given probability, it is natural to ask what is the *maximum* probability of satisfying that property, and in turn what control scheme produces the maximal value. Probabilistic reachability analysis is therefore naturally expressed as a *stochastic* optimal control problem, with the added benefit of controller synthesis, as will be shown in Section 2.4. First, we define three relevant reachability problems.

Problem 2.1. *Given a controlled DTSHS $\mathcal{H}$ (Def.2.1), a compact Borel set $K \subseteq \mathcal{X} \times \mathcal{Q}$, an initial state $s_0 \in K$, and a time horizon N, we would like to find the maximum probability that state s_n stays within K for all $n = 0, \dots, N$:*

$$p^N_{\text{viab}}(s_0; K) = \sup_{\pi \in \Pi} \mathbb{P}^\pi \left[s_1 \in K, \dots, s_N \in K | s_0 \right]. \tag{2.2}$$

We may also wish to find the set of all $s_0 \in K$ such that $p^N_{\text{viab}}(s_0; K) \geq 1 - \alpha$, known as the probabilistic viable set:

$$\text{Viab}^N(\alpha; K) = \left\{ s_0 \in K : \, p^N_{\text{viab}}(s_0; K) \geq 1 - \alpha \right\}. \tag{2.3}$$

Problem 2.2. *Given a controlled DTSHS $\mathcal{H}$ (Def.2.1), a compact Borel set $T \subseteq \mathcal{X} \times \mathcal{Q}$, an initial state $s_0 \in \mathcal{S}$, and a time horizon N, we would like to find the maximum probability that state s_n reaches T for some $n \in [0, N]$:*

$$p^N_{\text{reach}}(s_0; T) = \sup_{\pi \in \Pi} \mathbb{P}^\pi \left[\exists \, n \in [0, N] \ \textit{s.t.} \ s_n \in T | s_0 \right]. \tag{2.4}$$

We may also wish to find the set of all $s_0 \in \mathcal{S}$ such that $p^N_{\text{reach}}(s_0; T) \geq 1 - \alpha$, known as the probabilistic reachable set:

$$\text{Reach}^N(\alpha; T) = \left\{ s_0 \in \mathcal{S} : \, p^N_{\text{reach}}(s_0; T) \geq 1 - \alpha \right\}. \tag{2.5}$$

Problem 2.3. *Given a controlled DTSHS $\mathcal{H}$ (Def.2.1), the compact Borel sets $K \subseteq \mathcal{X} \times \mathcal{Q}$ and $T \subseteq \mathcal{X} \times \mathcal{Q}$, an initial state $s_0 \in K$, and a time horizon N, we would like to find the maximum probability thst state s_n reaches T for some $n \in [0, N]$, and stays within K until time n:*

$$\begin{aligned} p^N_{\text{ra}}(s_0; K, T) = \sup_{\pi \in \Pi} \mathbb{P}^\pi \big[& \exists \, n \in [0, N] \ \textit{s.t.} \ s_n \in T, \\ & \textit{and} \ \forall \, i = 1, \dots, n-1, \ s_i \in K | s_0 \big]. \end{aligned} \tag{2.6}$$

We may also wish to find the set of all $s_0 \in K$ such that $p^N_{\text{ra}}(s_0; K, T) \geq 1 - \alpha$, known as the probabilistic reach-avoid set:

$$\text{RA}^N(\alpha; K, T) = \left\{ s_0 \in \mathcal{S} : \, p^N_{\text{ra}}(s_0; K, T) \geq 1 - \alpha \right\}. \tag{2.7}$$

The above three problems can be posed in a related fashion. For instance, Problem 2.2 is the dual to Problem 2.1 [TMKA13]. Letting $p^N_{\text{viab}}(\pi, s_0; K)$ and $p^N_{\text{reach}}(\pi, s_0; T)$ be the viability and reachability probabilities for a given policy π, respectively, the reachability probability can be expressed in terms of a viability probability.

$$p^N_{\text{reach}}(s_0; T) = 1 - \inf_{\pi \in \Pi} p^N_{\text{viab}}(\pi, s_0; T^c) \tag{2.8}$$

and similarly

$$p^N_{\text{viab}}(s_0; K) = 1 - \inf_{\pi \in \Pi} p^N_{\text{reach}}(\pi, s_0; K^c). \tag{2.9}$$

The reach-avoid set is inherently a subset of either the reachable or viable sets, because the probability of meeting both specifications will always be bounded above by the probability of meeting only one of those specifications. In addition, the reachability problem can be formulated as a reach-avoid problem with an empty avoid set ($K = \mathcal{S}$), so that the objective is to reach a target set while remaining within the entire state space $\mathcal{S}$, which is trivially satisfied.

Dependent on the type of solution strategy employed, the existence of a solution for one of the above problems implies the ability to solve the other problems, although which formulation (reachability, viability, or reach-avoid) is easiest to consider from a computational standpoint will be problem specific, and dependent on the structure of the target or safe set being considered.

2.3 Markov Decision Processes

Problems 2.1 - 2.3 can be formulated as an optimal control problem, in a similar manner to the optimal control of Markov decision processes (MDPs). As shown in [APLS08] and [SL10], the reachability/viability problem and reach-avoid problem, respectively, can be formulated as stochastic optimal control problems and solved using dynamic programming, similar to how MDPs are formulated and solved [Put05]. We therefore first provide an overview of MDPs, to clarify the relation between reachability analysis of a DTSHS and optimal control of an MDP.

Definition 2.3. *(Markov Decision Process $\mathcal{J}$) An MDP is a tuple $J = (\mathcal{S},\mathcal{U},T,R)$ where*

1. *$\mathcal{S}$ is a finite set of states*
2. *$\mathcal{U}$ is a finite set of control inputs*
3. *$T : \mathcal{S} \times \mathcal{S} \times \mathcal{U} \to [0,1]$ is a state transition function assigning a probability distribution to state s_{n+1} given state s_n and action u_n for all n: $T(s_{n+1}|s_n,u_n)$*
4. *$R : \mathcal{S} \times \mathcal{U} \to \mathbb{R}$ is a function assigning a reward at each time step n, given the current state s_n and action u_n, $R(s_n,u_n)$*

For the system to be Markov, the state s_n at time n must include enough information to make optimal decisions, i.e. the optimal control policy is Markov [SPK13] (Definition 2.2). The transition function T also must assign a probability to being in state s_n conditioned only on the previous state s_{n-1} and control input u_{n-1}. Additional information does not change the probability [Res02], so that

$$T(s_n|s_{n-1},\ldots,s_0,u_{n-1},\ldots,u_0) = T(s_n|s_{n-1},u_{n-1}). \tag{2.10}$$

Generally, when considering the optimal control of an MDP, the control objective is to maximize a reward (or cost) function expressed as the expected value of a sum of rewards $R(s_n,u_n)$, accrued at each time step n, as a function of the state and control input, i.e.

$$\max_{\pi\in\Pi} \mathbb{E}\left[\sum_{n=0}^{N} R\left(s_n,\mu_n(s_n)\right)\right]. \tag{2.11}$$

The linearity property of expected values allows (2.11) to be broken down sequentially. Minimizing the multi-stage problem (2.11) reduces to a series of one stage optimization problems.

This simplification comes from *Bellman's Principle*, which states that any optimal policy over a time horizon $[0,N]$ will necessarily be optimal over any sub-horizon $[n_1,n_2] \subseteq [0,N]$. In other words, whatever the decision at time $n_1 - 1$, the policy over $[n_1, n_2]$ will be optimal given the state resulting from the decision at time $n_1 - 1$ [Pow11].

Bellman's principle allows the cost function to be broken down into a series of value functions, representing the "cost-to-go" starting in state s_n and acting according to the optimal policy until the terminal time N. For cost function (2.11), the value function at time n is recursively defined in terms of the value function at time $n+1$.

$$V_n^*(s_n) = \max_{u \in \mathcal{U}} \left\{ R(s_n, u_n) + \gamma \sum_{s' \in \mathcal{S}} V_{n+1}(s') \tau_s(s'|s, u) \right\} \tag{2.12}$$

The optimal cost over horizon N is therefore computed backwards in time. The value function is first calculated at time N, and control input u_N is determined, for each state s_N. Then V_N^* is used to compute V_{N-1}^*, etc. The ability to break down the optimal control problem into a sequence of smaller subproblems is known as dynamic programming. The procedure of iteratively solving (2.12) is called value iteration.

The policy is implicitly defined as a look-up table that assigns an optimal control input at each time step according to

$$\mu_n(s_n) = \arg \max_{u \in \mathcal{U}} V_n^*(s_n). \tag{2.13}$$

The maximum operator "max" is used in place of the supremum operator "sup" because for $\mathcal{U}$ finite, the maximum is attained. For an infinite time horizon $N = \infty$, the subscript n in (2.12) and (2.13) is dropped. The value function (2.12) is iteratively solved until some convergence criteria is met, such as $\|V_n - V_{n-1}\| < \epsilon$. The infinite reward for starting in any state s_0 is then $V^*(s_0)$, with V^* the final value function once the convergence criteria is met. The optimal policy is stationary, and given by $\mu^*(s) = \arg \max_{u \in \mathcal{U}} V^*(s)$.

2.4 Stochastic Optimal Control Formulations

Although similar in nature to an MDP, the stochastic optimal control formulations for Problems 2.1 - 2.3 have more complex cost functions. The viability problem, as shown in [APLS08], actually has a multiplicative rather than additive cost function structure, and the reach-avoid problem has a sum-multiplicative cost function [SL10].

The probability (2.2) of Problem 2.1 can be equivalently expressed as an expected value:

$$p^N_{\text{viab}}(s_0; K) \sup_{\pi \in \Pi} \mathbb{E}^\pi \left[\prod_{n=0}^{N} 1_K(s_n) \middle| s_0 \right] \tag{2.14}$$

with $\mathbb{E}^\pi$ the expected value taken with respect to the probability measure $\mathbb{P}^\pi$ induced by policy π. The notation 1_K denotes the indicator function over set K, with

$$1_K(s) = \begin{cases} 1, & \text{if } s \in K \\ 0, & \text{else} \end{cases} . \tag{2.15}$$

The equivalence arises because $\mathbb{P}[s \in K] = \mathbb{E}[1_K(s)]$ for any set K and random variable s. This property is well-known and easy to verify. The property can be more generally stated as: For an event B, the probability that B occurs is equal to the expected value of the indicator function over event B.

For the viability problem, the event of interest is remaining within K for N time steps. Writing it as a product of indicator functions gives

$$\prod_{n=0}^{N} 1_K(s_n) = \begin{cases} 1, & \text{if } s_n \in K \,\forall\, n = 0 \ldots, N \\ 0, & \text{otherwise} \end{cases} . \tag{2.16}$$

The probability (2.2) is hence equal to the expected value over the product of indicator functions.

Skipping to the reach-avoid problem 2.3 (reachability will follow directly from the reach-avoid formulation), we can derive its cost function in a similar manner to viability. Again, indicator functions are used to express the probability (2.6) as an expected value.

In this case, the event of reaching set T while remaining within set K is given as

$$\sum_{n=0}^{N} \left(\prod_{i=0}^{n-1} 1_{K \setminus T}(s_i) \right) 1_T(s_n) = \begin{cases} 1, & \text{if } \exists\, n \text{ s.t. } s_n \in T \text{ and } s_i \in K \,\forall\, i = 0, \ldots, n-1 \\ 0, & \text{otherwise} \end{cases} .$$

(2.17)

Then (2.6) is equal to

$$p_{\text{ra}}^{N}(s_0; K, T) = \sup_{\pi \in \Pi} \mathbb{E}^{\pi} \left[\sum_{n=0}^{N} \left(\prod_{i=0}^{n-1} 1_{K \setminus T}(s_i) \right) 1_T(s_n) \middle| s_0 \right] \tag{2.18}$$

The reachability probability (2.4) is equal to (2.18) with $K = \mathcal{S}$, or to the viability probability (2.14) with respect to T^c using the infimum rather than supremum over all policies, as in (2.8).

Both (2.14) and (2.18) have a value function representation and can therefore be solved using dynamic programming. The value function for the viability probability is recursively defined as

$$\begin{aligned} V_{N,\text{viab}}^{*}(s_N) &= 1_K(s_N) \\ V_{n,\text{viab}}^{*}(s_n) &= \sup_{u \in \mathcal{U}} \left\{ 1_K(s_n) \int_{\mathcal{S}} V_{n+1,\text{viab}}^{*}(s_{n+1}) \tau_s(ds_{n+1} | s_n, u) \right\} \end{aligned} \tag{2.19}$$

so that $p_{\text{viab}}^{N}(s_0; K) = V_0^*(s_0)$. The optimal policy π^* for Problem 2.1 is then $\pi^* = (\mu_0^*, \ldots, \mu_{N-1}^*)$ with

$$\mu_n^*(s) = \arg\sup_{u \in \mathcal{U}} 1_K(s) \int_{\mathcal{S}} V_{n+1}^*(s_{n+1}) \tau_s(ds_{n+1} | s, u) \tag{2.20}$$

for all n.

Similarly for (2.18), the value function is given by

$$\begin{aligned} V_{N,\text{ra}}^{*}(s_N) &= 1_T(s_N) \\ V_{n,\text{ra}}^{*}(s_n) &= \sup_{u \in \mathcal{U}} \left\{ 1_T(s_n) + 1_{K \setminus T}(s_n) \int_{\mathcal{S}} V_{n+1,\text{ra}}^{*}(s_{n+1}) \tau_s(ds_{n+1} | s_n, u) \right\} \end{aligned} \tag{2.21}$$

The advantage to the optimal control formulation is that it not only computes probabilistic viable and reachable sets, but that it also synthesizes a controller designed to maximize the given objective. The disadvantage is that to feasibly solve Problems 2.1 - 2.3 using the value function representations (2.19) and (2.21), i.e. using dynamic programming and value iteration, the state space $\mathcal{S}$ should be finite and small. If not, the value function and policy must be generated for a large number of states. In particular, for $\mathcal{S}$ hybrid, the continuous space $\mathcal{X}$ must be discretized [AAP+07]. In higher dimensions, the discretization leads to prohibitive computation times, and a trade-off between accuracy and speed of obtaining a solution.

It is the shortcomings of the dynamic programming formulation that motivate the next chapter.

Chapter 3

Stochastic Reachability Analysis with Perfect Observations

This chapter examines two novel methods for the calculation of stochastic reachable sets. In particular, we examine a) particle (or scenario) approximations to expected values, and b) conversion of the reach-avoid probability to a chance-constrained convex optimization problem. Both methods allow for computation of the reach-avoid set in higher dimensions, as compared to other existing methods for computing stochastic reachable sets. This work was motivated by the desire to extend stochastic reachability verification techniques to spacecraft applications, and in particular to assessing performance of rendezvous maneuvers at close range. Numerical results for the particle and convex approximations to reachability probabilities are therefore presented for the spacecraft rendezvous problem.

3.1 Introduction

Current solution strategies for reachability analysis of stochastic systems (namely dynamic programming) do not extend well to higher dimensional systems because of the need to use a finite state abstraction, such as a discretization of the state space. The computational effort required for larger problems renders a solution unattainable.

Current efforts have been limited to at most three dimensional systems. The focus of this chapter is therefore on more efficient means of generating probabilistic reachable sets that do not require discretization of the state space. We examine in particular the stochastic reach-avoid problem 2.3 of Chapter 2 in order to analyze a spacecraft rendezvous and docking problem.

The ability to safely perform spacecraft rendezvous and docking maneuvers is crucial in many applications, for instance in repairing satellites, resupplying the space station, and other missions. The potential for damage or loss in such expensive systems, in combination with prohibitively long communication delay, means that autonomy must be accurate, reliable, and effective despite uncertainties in modeling and in disturbance forces. For spacecraft dynamics, there are many initial states from which no control input will lead to a safe or desirable outcome (i.e. successful docking or rendezvous). The risk of costly failures can be reduced by determining whether a rendezvous-type maneuver is initialized from a position where success is guaranteed, or is guaranteed with high probability.

The linearized discrete time Clohessy - Wiltshire - Hill (CWH) equations for spacecraft relative motion with added stochastic noise can be broken up into in-plane and out-of-plane motion, i.e. a four-dimensional problem in x, y, $\dot{x}$, $\dot{y}$ and a separate two-dimensional problem in z and $\dot{z}$, within the Hill reference frame [Wie89]. Unfortunately, even the four-dimensional problem is beyond the limits of standard dynamic programming. This has led us to explore alternative methods for calculating reach sets which are more tractable for larger scale problems.

The rest of the chapter is organized as follows. After presenting some of the related work upon which the results of this chapter are based, we describe the general problem formulation in Section 3.3. The two methods we use for reach-avoid calculations are given in Section 3.4. Both methods are applied to the spacecraft rendezvous problem in Section 3.5 and their performance is discussed. Finally some concluding remarks are given in Section 3.6

3.2 Related Work

This chapter draws upon existing literature for solving stochastic control problems with chance constraints in order to develop novel techniques for calculating stochastic reach-avoid sets for higher dimensional systems. A chance constrained optimal control problem requires minimizing some objective (such as minimizing the control effort) while enforcing a probabilistic constraint, such as remaining within a safe region K with some desired probability. We examine two approaches: The first is based on sampling from the distribution of the noise to approximate expected values and probabilities (referred to as a "particle" or "scenario" approximation, see [NS06]), and was used in chance-constrained predictive control in [BOBW10] amongst others. The second approach is based on a convex under-approximation of the probability of remaining within a set K, assuming the additive noise is Gaussian, that allows the reachability problem to be solved using convex optimization methods. This method was presented in [BO09] and used again in [VT11], and applied to systems in two dimensions. In both cases, we exploit the linearity of the dynamics when applying them to the reachability problem.

While there is extensive literature on control mechanisms for performing spacecraft rendezvous maneuvers, particularly the use of model predictive control with approach and docking constraints for successful rendezvous (see, e.g. [WKBE12], [PDCK11], [HTRM12], [GVC12]), almost no work has been done on characterizing the initial set of states from which such maneuvers can be performed safely.

3.3 Problem Formulation

The DTSHS formulation of Definition 2.1 is simplified to a single continuous state, so that $\mathcal{S} = \mathcal{X} = \mathbb{R}^m$. Rather than write s_n for the state of the system at time n, we use $\mathbf{x}_n$ to indicate that the state is not hybrid, and is a vector in $\mathbb{R}^m$ (to distinguish between Cartesian coordinate x in $\mathbb{R}$).

The dynamics that propagate the state are

$$\mathbf{x}_{n+1} = A\mathbf{x}_n + Bu_n + v_n \tag{3.1}$$

with $A \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{m\times l}$ and $u \in \mathbb{R}^l$. The Gaussian noise $v_n \in \mathbb{R}^m$ is assumed zero mean with covariance matrix $\mathcal{V}$. The space of control inputs $\mathcal{U}$ is assumed a hyper-rectangle of the form $\mathcal{U} = \{u \in \mathbb{R}^l : \|u\|_1 \leq u_{\max}\}$ for simplicity, although any convex $\mathcal{U}$ is appropriate.

The techniques we utilize to solve Problem 2.3 most naturally express the reach-avoid objective with a terminal reach condition, i.e. that give the probability of reaching target set T at final time N specifically. All subsequent derivations assume a terminal reach constraint, and the true reach-avoid probability of (2.6) can be approximated by combining the reach-avoid probabilities given terminal times $n = 0, \ldots, N$:

$$p^N_{\mathrm{ra}}(\mathbf{x}_0; K, T) \leq \sum_{n=0}^{N} \tilde{p}^n_{\mathrm{ra}}(\mathbf{x}_0; K, T) \tag{3.2}$$

with $\tilde{p}$ denoting the reach-avoid problem with a terminal reach constraint. The summation in (3.2) must of course be capped at one, so that the probability makes sense.

3.4 Calculating Reach-Avoid Probabilities

The linearity of the system allows us to rewrite the dynamics (3.1) in vector form, letting $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]^T$, $\mathbf{u} = [u_0, u_1, \cdots, u_{N-1}]^T$ and $\mathbf{v} = [v_0, v_1, \cdots, v_{N-1}]^T$. Then

$$\mathbf{x} = \overline{\mathbf{x}}_0 + H\mathbf{u} + G\mathbf{v} \tag{3.3}$$

with $\overline{\mathbf{x}}_0$, H, and G defined as in [SB10].

$$\overline{\mathbf{x}}_0 = \begin{bmatrix} A\mathbf{x}_0 \\ A^2\mathbf{x}_0 \\ \vdots \\ A^N\mathbf{x}_0 \end{bmatrix}, H = \begin{bmatrix} B & 0 & 0 & \cdots & 0 \\ AB & B & 0 & \cdots & 0 \\ A^2B & AB & B & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & \cdots & \cdots & \cdots & B \end{bmatrix},$$

$$G = \begin{bmatrix} I & 0 & \cdots & 0 \\ A & I & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1} & A^{N-2} & \cdots & I \end{bmatrix}$$

The problem of finding $\tilde{p}^N_{\mathrm{ra}}(\mathbf{x}_0; K, T)$ can be posed as a constrained stochastic optimization problem.

Problem 3.1. *(Stochastic Optimal Control Formulation)*

$$\max_{\mathbf{u}} \mathbb{E}^{\mathbf{u}} \left[\left(\prod_{n=0}^{N-1} 1_K(\mathbf{x}_n) \right) 1_T(\mathbf{x}_N) \right]$$

Subject to:

$$\mathbf{x} = \overline{\mathbf{x}}_0 + H\mathbf{u} + G\mathbf{v}$$

$$u_n \in \mathcal{U} \; \forall n = 0, \ldots, N-1$$

Problem 3.1 is expressed as a single optimization problem rather than a multistage one that the value function (2.21) is designed to solve. Therefore, it returns an *open-loop* control vector $\mathbf{u}$ rather than a feedback policy π.

3.4.1 Particle Approximation

When taking the expected value of a function of a random variable $f(\mathbf{x})$ with respect to some distribution $p(\mathbf{x})$, it is necessary to evaluate an integral of the form

$E[f(\mathbf{x})] = \int f(x)p(x)\,dx$. However, this integral is often difficult or impossible to solve (as is the case for (2.18), where the dimension of the integral grows quite large). Instead, it is possible to approximate the integral by drawing independent, identically distributed random samples (particles) $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}, \cdots$ from a *proposal* distribution $q(\mathbf{x})$ and calculating a weighted sample mean

$$\hat{\mathbb{E}}\left[f(\mathbf{x})\right] = \frac{1}{M}\sum_{i=1}^{M} w_i f(\mathbf{x}^{(i)}) \tag{3.4}$$

with $w_i = \frac{p(x^{(i)})}{q(x^{(i)})}$ and $q(\mathbf{x})$ chosen so that $q(\mathbf{x}) > 0$ whenever $p(\mathbf{x}) > 0$. By the strong law of large numbers (and so making some weak assumptions on the boundedness of $f(\mathbf{x})$ and the moments of $p(\mathbf{x})$), it follows that $\hat{\mathbb{E}}\left[f(\mathbf{x})\right] \to \mathbb{E}[f(\mathbf{x})]$ as $M \to \infty$.

Because we can easily sample from the multivariate Gaussian distribution, we let $q(\mathbf{x}) = p(\mathbf{x})$, where $p(\mathbf{x})$ is Gaussian, and so $w_i = 1$. We therefore draw M samples $\mathbf{v}^{(1)}$, $\mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(M)}$, where $\mathbf{v}^{(i)} \in R^{m\times N}$

$$\mathbf{v}^{(i)} \sim \mathcal{N}(0, \overline{\mathcal{V}}), \qquad \overline{\mathcal{V}} = \begin{bmatrix} \mathcal{V} & 0 & \cdots & 0 \\ 0 & \mathcal{V} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathcal{V} \end{bmatrix}$$

We then have M realizations of the dynamics, given by $\mathbf{x}^{(i)} = \overline{\mathbf{x}}_0 + H\mathbf{u} + G\mathbf{v}^{(i)}$, for $i = 1, \ldots, M$. To approximate the cost function of Problem 3.1 we create an indicator variable z_i, where

$$z_i = \begin{cases} 1 & \text{if } \mathbf{x}_1^{(i)}, \ldots \mathbf{x}_{N-1}^{(i)} \in K \text{ and } \mathbf{x}_N^{(i)} \in T \\ 0 & \text{else} \end{cases} \tag{3.5}$$

Then $\tilde{p}_{\mathrm{ra}}^{N}(\mathbf{x}_0; K, T) \approx \frac{1}{M}\sum_{i=1}^{M} z_i$. Next, if the sets K and T are assumed convex, we may exploit the property that convex sets can be represented by a finite intersection of hyperplanes (see [BOBW10]), i.e.

$$\mathbf{x}_{1:N-1} \in K \wedge \mathbf{x}_N \in T \implies \mathbf{x} \in \bigcap_{l} \left\{\mathbf{x} : a_l^T \mathbf{x} \leq b_l\right\} \tag{3.6}$$

To convert Problem 3.1 to a mixed integer linear program, as in [BOBW10], we enforce constraints of the form (3.6) by using z_i and (3.5), and some large number C

$$a_l^T \mathbf{x}^{(i)} - b_l \leq C(1 - z_i) \, \forall \, i = 1, \cdots, M, \;\; l = 1, \cdots, M_l \tag{3.7}$$

so that for appropriately defined a_l, b_l representing sets K and T, and C large enough, we force $z_i = 0$ when $\mathbf{x}_n^{(i)} \notin K$ for some $1 \leq n \leq N - 1$, or $\mathbf{x}_N^{(i)} \notin T$. We want to find an open loop control $\mathbf{u}$ that produces as many z_i equal to 1 as possible.

Problem 3.2. *(Particle Approximation to Problem 1)*

$$\max_{\mathbf{u}} \sum_{i=1}^{M} z_i$$

Subject to:

$$\begin{aligned} \mathbf{x}^{(i)} &= \overline{\mathbf{x}}_0 + H\mathbf{u} + G\mathbf{v}^{(i)} \quad \forall \, i = 1, \ldots, M \\ a_l^T \mathbf{x}^{(i)} - b_l &\leq C(1 - z_i) \quad \forall \, i = 1, \ldots, M, \, l = 1, \ldots, M_l \\ |u_n| &\leq u_{\max} \quad \forall \, n = 0, \ldots, N - 1 \\ z_i &\in \{0, \, 1\} \, \forall \, i = 1, \ldots, M \end{aligned}$$

Problem 3.2 has a large number of variables that grow as the number of particles M and the number of time steps N increase, and is therefore still limited in terms of how many particles can be used and the amount of time given to reach the target.

One advantage to the particle method is that accuracy can be traded for computation time. Fewer particles can lead to a quick approximation of the reach-avoid probability for different $\mathbf{x}_0$, and those $\mathbf{x}_0$ that seem to produce larger probabilities can be recalculated with more particles to obtain a more accurate result if desired.

3.4.2 Convex Chance-Constrained Approximation

An alternative to the particle approximation approach is to reformulate Problem 3.1 by moving $\tilde{p}_{\mathrm{ra}}^N(\mathbf{x}_0; K, T)$ from the objective function to a chance constraint that must be enforced with probability $1 - \alpha$.

Problem 3.3. *(Chance Constrained Formulation of Problem 1)*

$$\min_{\mathbf{u}} \alpha$$

Subject to:

$$\tilde{p}_{\text{ra}}^{N}(\mathbf{x}_0; K, T) \geq 1 - \alpha \tag{3.8}$$

$$|u_n| \leq u_{\max} \; \forall\, n = 0, \ldots, N-1$$

Generally, this would not make the problem any easier to solve, since we still have to evaluate $\tilde{p}_{\text{ra}}^{N}(\mathbf{x}_0; K, T)$. However, because of the linearity and Gaussian noise, we can actually approximately solve Problem 3.1 by breaking (3.8) into univariate Gaussian constraints, which are convex (see [BO09], [VT11]). To see this, use the converse of (3.6) and Boole's inequality to write

$$\begin{aligned} \mathbf{x}_{1:N-1} \notin K \vee \mathbf{x}_N \notin T \Rightarrow \mathbf{x} \in \bigcup_{l} \left\{ \mathbf{x} : \, a_l^T \mathbf{x} > b_l \right\} \Rightarrow& \\ \mathbb{P}[\mathbf{x}_{1:N-1} \notin K \vee \mathbf{x}_N \notin T] = \mathbb{P}[\bigcup_{l} \left\{ \mathbf{x} : a_l^T \mathbf{x} > b_l \right\}] \leq \sum_{l} \mathbb{P}[a_l^T \mathbf{x} > b_l]& \end{aligned} \tag{3.9}$$

The inequality in the above expression indicates an upper bound on the probability of *not* being in K (or T at time N), and therefore a lower bound on the actual reach-avoid probability, which is still desirable.

Noting that $a_l^T \mathbf{x}$ is a scalar, and $a_l^T \mathbf{x} = a_l^T(\overline{\mathbf{x}}_0 + H\mathbf{u} + G\mathbf{v})$, it follows that $a_l^T \mathbf{x} \sim \mathcal{N}(a_l^T(\overline{\mathbf{x}}_0 + H\mathbf{u}), a_l^T G \overline{\mathcal{V}} G^T a_l)$ has a univariate Gaussian distribution. Applying (3.9), constraint (3.8) can be rewritten as follows.

$$\begin{aligned} \tilde{p}_{\text{ra}}^{N}(\mathbf{x}_0; K, T) \geq 1 - \sum_{l=1}^{M_l} \mathbb{P}\left[a_l^T X > b_l\right] \geq 1 - \alpha& \\ \Rightarrow \sum_{l=1}^{M_l} \mathbb{P}\left[a_l^T X > b_l\right] \leq \alpha& \end{aligned}$$

By taking a "risk allocation" approach as in [BOBW10], we can allow for a different probability of violating each individual constraint, i.e. for each l, $\mathbb{P}\left[a_l^T X > b_l\right] \leq \alpha_l$, with $\sum \alpha_l = \alpha$. We then require each $\alpha_l \geq 0$ and $\sum \alpha_l = \alpha \leq 1$, leading to the following approximation to Problem 3.3.

Problem 3.4. *(Chance Constrained Convex Approximation to Problem 1)*

$$\min \sum_{l=1}^{M_l} \alpha_l$$

Subject to:

$$1 - \Phi\left(\frac{b_l - a_l^T\mathbf{x}}{a_l^T G\overline{\mathcal{V}}G^T a_l}\right) \leq \alpha_l \,\forall\, l = 1, \ldots, M_l$$

$$|u_n| \leq u_{\max} \,\forall\, n = 0, \ldots, N-1$$

$$\sum_{l=1}^{M_l} \alpha_l \leq 1$$

$$\alpha_l \geq 0 \,\forall\, l = 1, \ldots, M_l$$

The function $\Phi(\,\cdot\,)$ indicates the standard normal cumulative distribution function, which is a concave function so long as its argument is non-negative ($\Phi(x)$ is concave for all $x \geq 0$), implying that $1 - \Phi(x)$ is convex for all $x \geq 0$. Problem 3.4 is therefore convex so long as $b_l - a_l^T\mathbf{x} \geq 0$, or $\alpha_l \leq 0.5$ for all l. If any α_l is greater than 0.5, the reach-avoid probability for the given $\overline{\mathbf{x}}_0$ will be quite low, so that we will likely only be concerned with those instances of Problem 3.4 when the problem is indeed convex, and hence we are guaranteed a solution.

3.4.3 Particle Approximation Using Feedback

We now show how to calculate $\tilde{p}^N_{\mathrm{ra}}(\mathbf{x}_0; K, T)$ when $\mathbf{u} = W\mathbf{x} + \mathbf{u}_0$ with W block lower triangular, and u_n bounded between known maximum and minimum values, i.e. $\mathcal{U} = \{W, \mathbf{u}_0 : |W\mathbf{x} + \mathbf{u}_0| \leq \mathbf{u}_{\max}\}$ ($\mathbf{u}_{\max} = u_{\max} \times [1, \ldots, 1]$). We only consider the particle approximation method in the case of feedback, because the chance-constrained method will no longer be convex, and the constraint in 3.4 involving $\Phi\left(\frac{b_l - a_l^T\mathbf{x}}{a_l^T G\overline{\mathcal{V}}G^T a_l}\right)$ becomes difficult to enforce. Problem 3.2, however, can be modified nicely to accommodate a feedback controller.

First, the expression for $\mathbf{x}$ must be modified to incorporate feedback in (3.3). Both $\mathbf{x}$ and $\mathbf{u}$ can be written as affine functions of the random vector $\mathbf{v}$, as shown in

[SB10]:

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{u}\end{bmatrix} = PV + \begin{bmatrix}\tilde{\mathbf{x}}\\ \tilde{\mathbf{u}}\end{bmatrix}$$

$$P = \begin{bmatrix}G + HK(I - HW)^{-1}G\\ W(I - HW)^{-1}G\end{bmatrix}$$

$$\begin{bmatrix}\tilde{\mathbf{x}}\\ \tilde{\mathbf{u}}\end{bmatrix} = \begin{bmatrix}\overline{\mathbf{x}}_0 + H\mathbf{u}_0 + HW(I - HW)^{-1}(\overline{\mathbf{x}}_0 + H\mathbf{u}_0)\\ W(I - HW)^{-1}(\overline{\mathbf{x}}_0 + H\mathbf{u}_0) + \mathbf{u}_0\end{bmatrix}$$

Since these equations are not convex in the variables W and $\mathbf{u}_0$, new variables $Q = W(I - HW)^{-1}$ and $R = (I + QH)U_0$ are introduced to make the equations convex. The variables of interest, W and $\mathbf{u}_0$, can be recovered after solving for optimal Q and R, as in [SB10]. Note that $\mathbf{u}$ is random, because it is a function of the Gaussian vector $\mathbf{v}$. We therefore cannot impose with certainty constraints on the maximum and minimum values of $\mathbf{u}$. We instead require $\mathbb{E}[|\mathbf{u}|] \leq \mathbf{u}_{\max}$, which can also be evaluated in the manner of (3.4). Problem 3.2 is reformulated as:

Problem 3.5. *(Particle Approximation with Feedback)*

$$\max \sum_{i=1}^{M} z_i$$

Subject to:

$$\mathbf{x}^{(i)} = (I + HQ)GV^{(i)} + (I + HQ)\overline{\mathbf{x}}_0 + HR \;\; \forall\, i = 1, \ldots, M$$

$$\mathbf{u}^{(i)} = QGV^{(i)} + Q\overline{\mathbf{x}}_0 + R \;\; \forall\, i = 1, \ldots, M$$

$$a_l^T \mathbf{x}^{(i)} - b_l \leq C(1 - z_i) \;\; \forall\, i = 1, \ldots, M,\; l = 1, \ldots, M_l$$

$$\frac{1}{M}\sum_{i=1}^{M} |\mathbf{u}^{(i)}| \leq \mathbf{u}_{max}$$

$$z_i \in \{0,\, 1\} \;\; \forall\, i = 1, \ldots, M$$

Q block lower triangular

The number of variables and constraints significantly increases in comparison to Problem 3.2, as does the solution time.

3.5 Application to Spacecraft Rendezvous and Docking

We apply both of the open-loop methods (Problem 3.2 and Problem 3.4) to the reach-avoid problem for the final phase of in-plane space rendezvous, as described in [WKBE12].

The in-plane dynamics for an approaching spacecraft (the deputy) relative to its target (the chief) are given by the linearized time-invariant CWH equations [Wie89]:

$$\ddot{x} - 3\omega^2 x - 2\omega\dot{y} = \frac{F_x}{m_c} \tag{3.10}$$

$$\ddot{y} + 2\omega\dot{x} = \frac{F_y}{m_c} \tag{3.11}$$

Here F_x and F_y are the components of the external force vector (i.e. the thruster control input), m_c is the mass of the deputy, and $\omega = \sqrt{\frac{\mu}{R_0^3}}$ where μ is the gravitational constant and R_0 is the orbital radius of the spacecraft. In the following we will only consider the in-plane motion since the out-of-plane motion is decoupled from the x, y dynamics. We then discretize (3.10)-(3.11) using time-step Δ according to [AM07], and let $\mathbf{x} = [x,\, y,\, \dot{x},\, \dot{y}]^T$, $u = \left[F_x,\, F_y\right]^T$. We obtain discretized dynamics of the form (3.1) with

$$A = e^{\tilde{A}T}, \quad B = \int_0^T e^{\tilde{A}t}\tilde{B}\,dt$$

and

$$\tilde{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3\omega^2 & 0 & 0 & 2\omega \\ 0 & 0 & -2\omega & 0 \end{bmatrix} \quad \tilde{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m_c} & 0 \\ 0 & \frac{1}{m_c} \end{bmatrix}$$

with added Gaussian process noise vector $v_n \in \mathbb{R}^4$, that represents uncertainty in the model due to external forces on the spacecraft not captured in the linearized model.

The problem of controlling the deputy to approach the chief according to the dynamics (3.10) - (3.11) includes many safety constraints that must be satisfied for the operation to be successful. For instance, the spacecraft should come close to the chief without hitting it, while staying within a line-of-sight (LoS) cone relative to the chief. Staying within an LoS cone is necessary when there are sensors on the chief measuring the location of the approaching spacecraft. While we do not incorporate sensor measurements, it is a natural next step, and so we assume all necessary requirements as if sensor measurements were taken.

We define the safe set K that the deputy should remain within for all time steps 0 to $N-1$ as the LoS cone at the origin in x and y (unchanging over time), and a square in $\dot{x}$ and $\dot{y}$, i.e. the maximum and minimum velocities are bounded for all time steps. Target set T is defined as a box close to the chief that the deputy should reach at time N. The set T also includes bounds on the velocity components $\dot{X}_N$ and $\dot{Y}_N$ so that the deputy does not crash or dock with excess force. Fig. 3.1 shows the safe set K and target set T for states x and y.

We set covariance $\mathcal{V}$ as a diagonal matrix with entries $[1e-4, 1e-4, 5e-8, 5e-8]$. The noise covariance is kept small, since open-loop controllers do not deal well with noise over longer periods of time (and the covariance grows with N). As the covariance grows, the potential distance between different realizations of the trajectory $\mathbf{x}$ grows as well, and it is harder for one controller, without feedback, to drive all possible

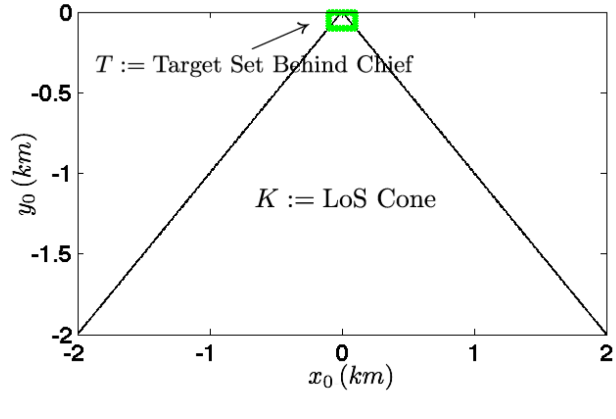


Figure 3.1: The sets K and T in the x and y dimensions. The black lines represent the LoS cone the deputy is trying to remain within, and the green box shows the target that is close to the chief (placed at the origin).

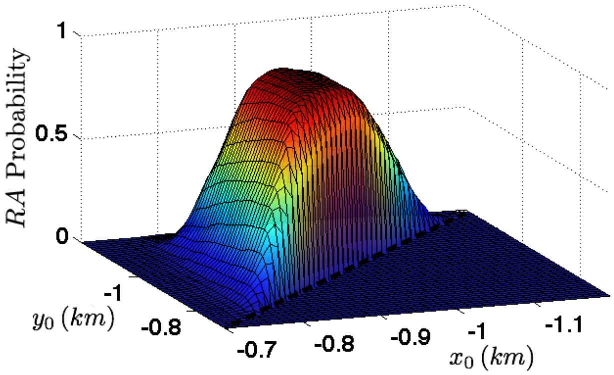
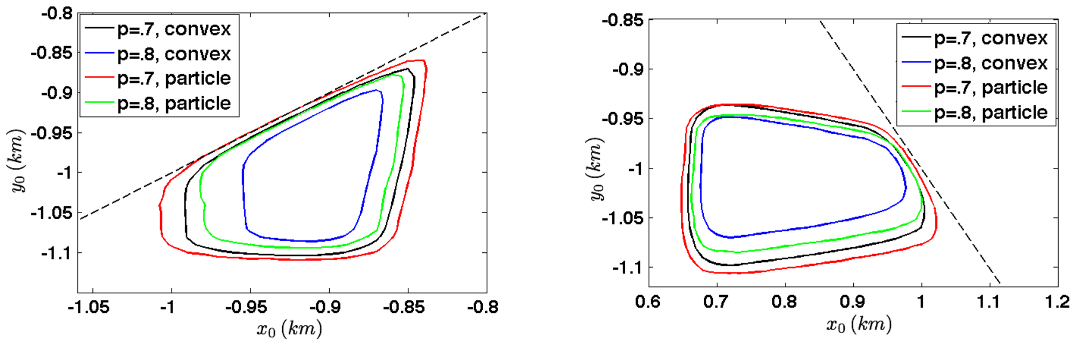


Figure 3.2: Reach-avoid probability starting at various x_0 and y_0, for positive initial velocities $\dot{x}_0 = 0.01$ km/s, $\dot{y}_0 = 0.01$ km/s, using the particle approximation method with 800 particles. The dotted black line indicates the boundary of the LoS cone stemming from the origin. The area of nonzero probability is contained in the region where x_0 and y_0 are both negative, because both initial velocities are positive.

realizations to the origin while keeping them within an LoS cone.

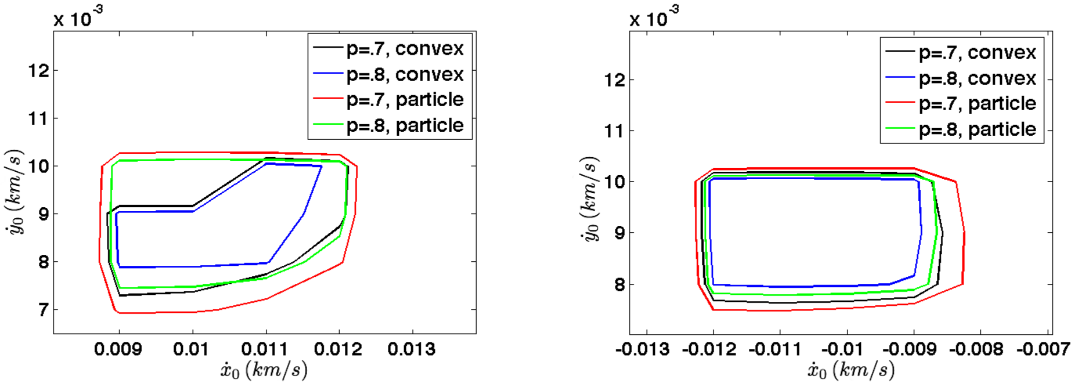
Problems 3.2 and 3.4 are solved over a mesh of initial $\mathbf{x}_0$ values ranging from 2 km behind the chief to directly behind it. We compute the reachable sets over a period of $N = 5$ time steps, $\Delta = 20$ seconds per time step (as in [WKBE12]), and compare the particle approach of Section 3.4.1 to the convex optimization approach of Section 3.4.2. Problem 3.2 is solved in CPLEX [IBM99] and Problem 3.4 using an active-set algorithm through *fmincon* in MATLAB's Optimization toolbox. We start with a coarse grid for $\mathbf{x}_0$, with x_0 equally spaced 0.1 km apart ranging from -2 to 2 km, y_0 ranging from -2 km to 0 km with the same spacing as x_0, and $\dot{x}_0$, $\dot{y}_0$ ranging from -0.2 km/s to 0.2 km/s, equally spaced 0.01 km/s apart. We then isolate the states $\mathbf{x}_0$ leading to higher reach-avoid probabilities, refine the mesh limited to these values, and recalculate the reach-avoid probabilities.

Fig. 3.2 shows the probability of staying within K and reaching T in 5 time steps, over varying x_0 and y_0 (using the refined mesh with 0.01 km between points), with fixed $\dot{x}_0 = \dot{y}_0 = 0.01$ km/s, using the particle approach with 800 particles. The area of non-zero probability is limited to where both x_0 and y_0 are negative, because the



(a) (b)

Figure 3.3: The sets $\mathrm{RA}^5(0.3; K, T)$ and $\mathrm{RA}^5_{0.2}(K, T)$ (i.e. the sets of all X_0 such that $\tilde{p}^5_{\mathrm{ra}}(\mathbf{x}_0; K, T) \geq 0.7$ and $\tilde{p}^5_{\mathrm{ra}}(\mathbf{x}_0; K, T) \geq 0.8$) with $\dot{x}_0 = 0.01$ km/s, $\dot{y}_0 = 0.01$ km/s in (a), and $\dot{x}_0 = -0.01$ km/s, $\dot{y}_0 = 0.01$ km/s in (b), fixed. The reach-avoid sets for $\alpha = 0.3$ and $\alpha = 0.2$ are shown for the convex approximation (black, blue), and for the particle approximation using 800 particles (red, green) methods. The dotted black line indicates the boundary of the LoS cone. RA sets are not symmetric when $\dot{x}_0$ switches from positive to negative because of the asymmetric nature of the dynamics.



(a) (b)

Figure 3.4: The sets $\mathrm{RA}^5(0.3; K, T)$ and $\mathrm{RA}^5(0.2; K, T)$ (i.e. the sets of all $\mathbf{x}_0$ such that $\tilde{p}^5_{\mathrm{ra}}(\mathbf{x}_0; K, T) \geq 0.7$ and $\tilde{p}^5_{\mathrm{ra}}(\mathbf{x}_0; K, T) \geq 0.8$) with $x_0 = -0.9$ km, $y_0 = -1$ km in (a), and $x_0 = 0.9$ km, $y_0 = -1$ km in (b), fixed. The reach-avoid sets for $\alpha = 0.3$ and $\alpha = 0.2$ are shown for the convex approximation (black, blue), and for the particle approximation using 800 particles (red, green) methods. In both figures, the *RA* sets span only a few m/s in each direction, demonstrating the importance of accuracy in the initial velocities for rendezvous to be successful.

initial velocities are both positive, and over the time period of $N = 5$, the controller cannot sufficiently reverse the velocity of the deputy to enable it to still reach the origin. Further, the x_0 and y_0 leading to positive probability only lie in a range of approximately -1.1 km to -0.8 km, somewhat in the middle of the LoS cone. This is because we have restricted the deputy to rendezvous with the chief at *exactly* $N = 5$ time steps. Starting too far away, the controller cannot bring the deputy to the chief within the small time frame, and starting too close, the controller cannot prevent the deputy from overshooting the chief, hence the small region with positive probability of success. Taking the union over N of all reach-avoid sets with $N \leq 5$, would expand the full reach-avoid set of Fig. 3.2. Finally, there is no initial position leading to a reach-avoid probability higher than 0.9, even with the choice of a small noise covariance. Without feedback, the reach-avoid probability will grow smaller as N increases, which is already evident in just over 5 time-steps.

Fig. 3.3a shows specific reach-avoid probabilities (for a given probability $1 - \alpha$), computed with the particle and convex approaches. The set of all initial states x_0, y_0 leading to trajectories that will stay within K and reach T at $N = 5$, with probability 0.7 and 0.8 when $\dot{x}_0 = \dot{y}_0 = 0.01$ km/s, is shown. The sets generated by the particle approach (using 800 particles) are slightly larger than those from the convex approach, consistent with the fact that the convex approach gives a slight under-approximation of the reach-avoid set. A comparable scenario, with $\dot{x}_0 = -0.01$ km/s, is shown in Fig. 3.3b. Note in this case that the reach-avoid sets occur in the region where x_0 is positive, for the same reason that they appear where x_0 is negative when the initial x-velocity is positive.

The reach-avoid sets are not symmetric, because the zero-input dynamics also are not symmetric. Plotting sample trajectories of the zero-input, zero-noise CWH equations demonstrates this. Fig. 3.5 shows $\mathrm{RA}^5(0.2; K, T)$ for $\dot{y}_0 = 0.01$ km/s, computed with the particle approach. As $\dot{x}_0$ decreases, the size of the reach-avoid set shrinks in the y-direction, as in Fig. 3.3. The scale of the axis shows that the range of $\dot{x}_0$ for a given x_0 and y_0 is in fact much smaller relative to the ranges of x_0 and y_0 when $\dot{x}_0$ is fixed. This indicates that the reach-avoid probability is much more

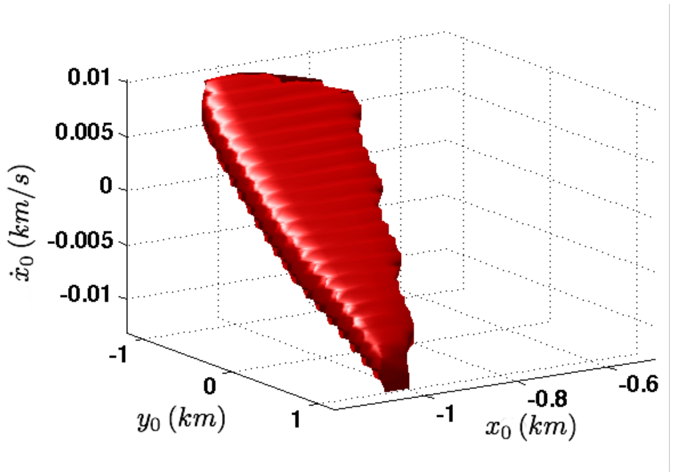


Figure 3.5: Three-dimensional reach-avoid set $\mathrm{RA}^5(0.2; K, T)$, with $\dot{y}_0$ fixed at 10 m/s, generated using the particle approximation method. As x_0 ranges from negative to positive, $\dot{x}_0$ ranges from positive to negative. The cross-sections for fixed $\dot{x}_0$ shrink as x_0 becomes positive due to the asymmetric nature of the zero-input dynamics. The difference in cross-sections for x_0 positive versus negative are highlighted in Fig. 3.3.

sensitive to the initial velocity than it is to the initial position. Fig. 3.4 explores this further.

Reach-avoid sets for fixed x_0, y_0 and varying $\dot{x}_0$, $\dot{y}_0$ using a mesh with 1 m/s between points using both the convex and particle approaches are given in Fig. 3.4. In Fig. 3.4a, $x_0 = -0.9$ km and $y_0 = -1$ km, and in Fig. 3.4b, x_0 is $+0.9$ km. The reach-avoid sets using the particle approach are approximately symmetric across the two values of x_0. The convex reach-avoid sets in Fig. 3.4a are slightly misshapen relative to all the other reach-avoid sets, which is unfortunately caused by the MATLAB algorithm, which is unable to find an optimal solution to Problem 3.4 for various initial guesses. Regardless, it is clear that for a given starting position x_0, y_0, the initial velocities must be specified to within an accuracy of a couple m/s, or the probability of success drops drastically, and in fact will drop to zero very quickly, at least in the case of an open loop controller.

To compare the performance of the open-loop controller to one that uses feedback, we next present results for the reach-avoid set calculated by solving Problem 3.5, in

Fig. 3.6. We fix $\dot{x}_0 = \dot{y}_0 = 0.01$ km/s in Fig. 3.6a, and $x_0 = -0.9$ km, $y_0 = -1$ km in Fig. 3.6b. All parameters (i.e. Σ and N) are kept the same as in the open-loop case, although the number of particles is reduced to 500, and a slightly coarser grid is used, in order to decrease computation time. A feedback controller drastically increases the size of the reach-avoid set, because it can steer those trajectories driven away from the origin by the process noise back to the desired path. This contrast in results between controllers highlights the importance of considering the type and effect of the controller to avoid overly conservative reachability results, and merits further discussion. It is therefore an area of potential future work.

Finally, we briefly compare the performance of both methods used to produce the above results in the open-loop case. Neither method is perfect, and both produce approximations to the true reach-avoid probabilities. Figs. 3.3 and 3.4 showed that the convex chance-constrained approach consistently under-approximates (as expected) the reach-avoid probabilities relative to the particle approximations, although never to the point of being unreasonable. We found that the particle method was consistent in its results, and has the advantage of allowing for a quick, coarse approximation using less particles, to then narrow down the region where the reach-avoid probabilities should be calculated more accurately. Both methods have the disadvantage that, if the reach-avoid probability for a given initial point is very small or zero, they will possibly never converge to a solution (or take an unreasonable amount of time), particularly in the case of Problem 3.4, since the problem is in fact no longer convex. Hence we found the need to stop the algorithm after it had taken sufficient time to find an optimal solution, were one to exist.

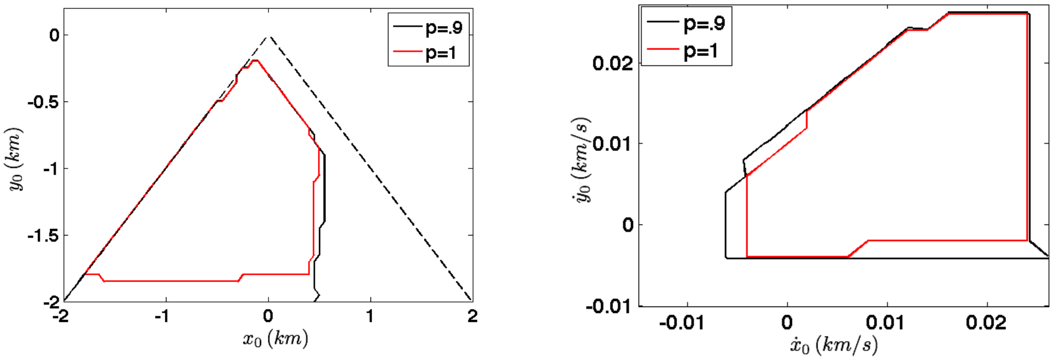
Computation times for the particle method depend heavily on the amount of particles used, which affects both the number of variables and constraints making up the problem. For instance, using 500 particles over $N = 5$ time steps, there are a total of 510 variables: 500 indicator variables z_i, one for each particle, and 10 variables for U, since $\mathbf{u} \in \mathbb{R}^{10}$. Further, to define inequality constraints ensuring $\mathbf{x}_1, \ldots, \mathbf{x}_4 \in K$ and $X_5 \in T$, we need $M_l = 48$ (a_l, b_l) pairs of inequalities, producing $500 \times 48 = 24000$ constraints of the form $a_l^T \mathbf{x}^{(i)} - b_l \leq C(1 - z_i)$. However, using commercially available

MILP solvers, such as CPLEX, for a single given $\mathbf{x}_0$ this problem was solved in 6.88 seconds on a 2 GHz Macbook with 2 GB of RAM.

Compared to the particle approach, Problem 3.3 has far fewer variables and constraints. For the same case with $N = 5$ time steps, we again have $M_l = 48$ constraints $a_l^T X \leq b_l$ confining $\mathbf{x}_1, \ldots, \mathbf{x}_4 \in K$ and $\mathbf{x}_5 \in T$. Associated to each constraint is a variable α_l, so that there are $48 + 10 = 58$ variables total (and again the 10 variables correspond to $\mathbf{u} \in \mathbb{R}^{10}$). The drawback here is the nonlinear constraint involving the Gaussian normal density function, $1 - \Phi\left(\frac{b_l - a_l^T \mathbf{x}}{a_l^T G\bar{\mathcal{V}}G^T a_l}\right) \leq \alpha_l$. Nonlinear solvers can be more time consuming if the initial solution point provided to the algorithm is far from the optimal solution. For an initial solution generated randomly, and the same $\mathbf{x}_0$ as was implemented in the particle method, Problem 3.4 was solved in 2.81 seconds using MATLAB's *fmincon* function and its active-set algorithm. However, for some initial points the algorithm can fail to converge completely and the process must be repeated with a different initial point, adding to the overall time of the algorithm. With the same $\mathbf{x}_0$ but a different initial point, the problem was solved in 50.99 seconds with 7 different initial points before finding the solution. Using different algorithms or software and "good" initial guesses may help speed up the algorithm.

The convex method was generally faster, but only when the initial values over which the reach-avoid sets were calculated had been narrowed down enough to eliminate most of the low probability calculations, and when "good enough" initial solutions were provided to avoid recalculating the reach-avoid probability several times.

The data for Figs. 3.3 and 3.4 generally took about 20 minutes each to calculate (with no significant difference in time between the methods). However, the particle approach was used to generate Fig. 3.2 and Fig. 3.5 because over a wide range of values it was found to be faster. The particle approximation does not get "stuck" at certain initial states $\mathbf{x}_0$ like the convex approximation, when computation takes much longer (e.g. a 6.88 second computation for the particle approximation versus a 50.99 second computation for the convex approximation).



(a) (b)

Figure 3.6: The sets $RA^5_{0.1}(K, T)$ (black) and $RA^5_0(K, T)$ (red) (i.e. the sets of all X_0 such that $p^5_{X_0}(K, T) \geq 0.9$ and $p^5_{X_0}(K, T) = 1$) with $\dot{x}_0 = 0.01$ km/s, $\dot{y}_0 = 0.01$ km/s in (a), and $x_0 = -0.9$ km, $y_0 = -1$ km in (b), when using state feedback, calculated using the particle method with 500 particles. The dotted black line in (a) shows the LoS cone. The RA sets are much larger, and attain greater probabilities, than in the case of open loop control, spanning more than half of the LoS cone in (a) and a much larger range of velocities in (b).

While the results presented are for a four-dimensional system, both approaches should be suitable for reachability calculations of higher-dimensional systems as well. The number of constraints in either Problem 3.2 or 3.4 will not increase for higher dimensional systems (except possibly in the number of control inputs u to be bounded, which is insignificant compared to the other constraints). The main computation will lie in calculating the actual reach-avoid set, where the problem must be solved repeatedly over a much larger grid of $\mathbf{x}_0$ values. Again, the process can be sped up by solving over a coarse grid, and in the particle approximation case using a small number of particles, and then focusing and refining the grid to where the success probabilities are estimated to be highest.

Further, both methods can be used for reachability calculations of any linear system with additive Gaussian noise. The convex chance-constrained approach of Problem 3.4 will only work with Gaussian noise, but the particle method could be applied to noise following any known distribution. We required the reach and the complement of the avoid set to be convex, but nonconvex regions could also be addressed by decomposition into multiple convex regions (see, e.g. [OBW10]).

3.6 Summary

Two novel methods for reachability calculations on higher-dimensional linear stochastic systems have been presented. One involves sampling from the noise distribution to generate a particle approximation to the expected value in Problem 3.1, which would otherwise be too difficult to evaluate, and using mixed integer linear programming to find $\tilde{p}^N_{\mathrm{ra}}(\mathbf{x}_0; K, T)$. The other exploits the linearity of the dynamics and the Gaussian assumption on the noise to produce a convex optimization problem also approximating $\tilde{p}^N_{\mathrm{ra}}(\mathbf{x}_0; K, T)$, by making the reach-avoid probability a chance constraint whose bound is actually the objective to be maximized. An open-loop control vector $\mathbf{u}$ was found to maximize the reach-avoid probability in both cases.

The obvious setback is that with the propagation of the noise, an open-loop controller cannot optimally control *all* realizations of the dynamics over an extended length of time. The noise leads to dynamics which diverge over time, so that the chances of reaching the target set T using an open-loop controller go to zero very quickly as the time horizon is extended.

We were able to impose a state-based feedback controller using the particle approximation method, despite the increasing number of variables and constraints. The natural next step is then to look more closely at closed-loop controllers. While the particle approximation method still works, the convex approximation technique breaks down when feedback is introduced, because the constraints are no longer convex. One possible approach to maintain convexity would be to use a model predictive controller to simulate feedback (although it may not help the speed of the algorithm), to see how this improves the stochastic reach-avoid probability.

Chapter 4

Stochastic Reachability Analysis with Partial Observations

We now focus on the reachability problem for stochastic hybrid systems with incomplete state information. We establish the theoretical foundations for both generating probabilistic reachable and viable sets and synthesizing controllers to maximize the likelihood of meeting such specifications. Our approach utilizes a sufficient statistic that reduces the problem to one of perfect state information. We develop a dynamic programming recursion for the solution of the equivalent perfect information problem, and prove that the recursion is valid, an optimal solution exists, and results in the same solution as to the original problem. A simplified dynamic programming recursion is also given, through the introduction of a change of measure to the sufficient statistic.

4.1 Introduction

The goal of this chapter is to extend the stochastic optimal control formulations for reachability presented in Chapter 2 to the case of a partially observable system, in which the controller only has access to noisy or incomplete measurements of the state. We examine the viability problem, and then extend those results to the reachability and reach-avoid problem. Incomplete state information requires a controller designed to account for the lack of full knowledge of the state. Ignoring the need for a controller tailored to the incomplete information problem could ultimately lead to suboptimal control inputs as compared to the case in which the true state of the system is known. For safety verification and reachable set calculations, if the controller is falsely assumed to have perfect knowledge of the state, the reachability probability may be overestimated. This is certainly undesirable in the context of safety verification.

Optimal control in the presence of a stochastic observation process can be directly related to a partially observable Markov decision process (POMDP). POMDPs provide a generalized modeling framework for a stochastic system whose state obeys the Markov property and is subject to a control input. The control input is often chosen optimally to maximize the expected value of a sum of rewards, such that at each time step some reward is accrued. The optimal control and maximum expected reward are found by using a sufficient statistic, which encapsulates and condenses all necessary information for control of the system. The sufficient statistic produces an alternate state over which the POMDP is defined, and which is known completely. For an additive cost (or reward) function, the posterior distribution, or probability density of the state given all available information (observations, control inputs) up to the present, provides sufficient information for control of the system (this result is derived in, e.g., [BS96]).

For a non-additive cost function the posterior distribution is no longer sufficient (see [Shi64]). A different sufficient statistic for reducing the problem to one of perfect information must be derived, which is the main focus of this chapter. We derive a) sufficient statistics for the multiplicative viability, reachability, and reach-avoid cost functions with hybrid state dynamics, and b) the dynamic programming equations to solve each problem in terms of the sufficient statistic. We also introduce a change of measure to the hybrid space, so that the observations are independent of the state of the system (and are in fact independent and identically distributed). This makes for simpler dynamic programming equations, and should aid in computation and simulations. Our main focus is on the theoretical foundations for solving the reachability problem for a partially observable DTSHS. We note that implementation of our technique requires further work in approximation strategies as well as in special classes of systems for which exact solutions are available. This is the subject of Chapters 5 and 6.

The chapter is organized as follows. First, we provide an overview of some of the related literature on verification of partially observable systems. We then extend the definition of a DTSHS to one with partial observations in 4.3, and modify the optimal control formulation for the viability problem (see 2.1) to reflect the limited information available to the controller. We then give a detailed overview of POMDPs in Section 4.4, because our approach directly relates to the optimal control of POMDPs. A sufficient statistic is derived for the viability problem in Section 4.5.1, and the equivalence of the original problem to one with perfect information expressed in terms of the sufficient statistic is established in Section 4.5.2. Alternate sufficient statistics that 1) incorporate a change of measure, and 2) are appropriate for reachability and reach-avoid cost functions, are given in Sections 4.5.3 and 4.5.4, respectively. In Section 4.6, we describe two examples of partially observable discrete time stochastic hybrid systems, demonstrate how to reformulate them in terms of our sufficient statistic, and discuss some of the computational challenges as well as possible solution strategies. Concluding remarks are given in Section 4.7.

4.2 Related Work

Until recently, verification of partially observable DTSHS had not been addressed. Concurrently with our efforts, however, [DAT13] considered safety specifications for partially observable DTSHS through a new optimal control formulation of the viability problem. Although viability was originally presented in terms of a multiplicative cost function ([APLS08]), [DAT13] rewrite it as a terminal cost function, by appending to the state of the hybrid system a binary variable representing whether the state has remained within the desired region up to the previous time step. This formulation produces a cost function that is additive, and the standard approach of re-framing the problem in terms of the belief state, and solving the equivalent perfectly observed problem, applies. Indeed, inspired by this approach, [TMKA13] reformulate the reachability problem more generally as an additive cost optimal control problem, although they do not discuss the partially observable case.

Although we preserve the multiplicative cost function, leading to a seemingly more complex problem, the additive cost formulation effectively moves the complexity from the cost function to the modified state of the system. The posterior distribution of the new state is actually the same as the distribution produced by the sufficient statistic we derive, so that ultimately the only advantage to the additive cost formulation is its familiar and well-studied form.

Our approach is inspired by similar work in risk-sensitive control problems [JBE94], [FGM97]. The risk-sensitive control problem minimizes the *exponential* of a sum of costs, rather than a sum of costs, so that the cost objective is in fact non-additive. In particular, [JBE94] derive a sufficient statistic for a partially observable discrete time nonlinear system, which is further analyzed and extended in the context of a POMDP in [FGM97]. In the latter, the state, observation, and control take values from finite, discrete sets, whereas in the former the state, observation, and control spaces are continuous. We also draw upon the theory from [BS96] to establish the definition of a sufficient statistic and validity of the dynamic programming equations.

4.3 Problem Formulation

4.3.1 Partially Observable Stochastic Hybrid System

We extend the DTSHS of Definition 2.1 in Chapter 2 to include an observation process, which we call a partially observable DTSHS (PODTSHS). Only the observation process is available to the controller, and is of the form $y_n = (y_n^x, y_n^q)$, where y_n^x is associated with x_n, and y_n^q with q_n. The observations are stochastic as well, and governed by independent stochastic kernels that are combined to produce a probability measure on the full observation process.

Definition 4.1. *(Partially Observable Discrete Time Stochastic Hybrid System $\mathcal{H}^{po}$). A PODTSHS is a tuple $\mathcal{H}^{po} = (\mathcal{X}, \mathcal{Q}, \mathcal{Y}^x, \mathcal{Y}^q, \mathcal{U}, T_x, T_q, T_{y^x}, T_{y^q}, \rho)$ where*

1. $\mathcal{X} \subseteq \mathbb{R}^n$ *is a set of continuous states*

2. $\mathcal{Q} = \{q_1, q_2, ...q_{N_q}\}$ *is a finite set of discrete states with cardinality* N_q*, with* $\mathcal{S} = \mathcal{X} \times \mathcal{Q}$ *the hybrid state space*

3. $\mathcal{Y}^x \subseteq \mathbb{R}^n$ *is a set of continuous observations*

4. $\mathcal{Y}^q \subseteq \mathcal{Q}$ *is a set of discrete observations, with* $\mathcal{Y} = \mathcal{Y}^x \times \mathcal{Y}^q$ *the hybrid observation space*

5. $\mathcal{U}$ *is a finite, bounded set of possible control inputs, affecting discrete and continuous state transitions*

6. $T_x : \mathcal{B}(\mathbb{R}^n) \times \mathcal{Q} \times \mathcal{S} \times \mathcal{U} \to [0, 1]$ *is a Borel-measurable stochastic kernel which assigns a probability measure to* x_{n+1} *given* $s_n = (x_n, q_n), u_n, q_{n+1}$ *for all n:* $T_x(dx_{n+1} \in B | q_{n+1}, s_n, u_n)$ *where* $B \in \mathcal{B}(\mathbb{R}^n)$*, the Borel* σ*-algebra on* $\mathbb{R}^n$

7. $T_q : \mathcal{Q} \times \mathcal{Q} \times \mathcal{U} \to [0, 1]$ *is a discrete transition kernel assigning a probability distribution to* q_{n+1} *given* q_n, u_n *for all n:* $T_q(q_{n+1}|q_n, u_n)$

8. $T_{y^x} : \mathcal{B}(\mathbb{R}^n) \times \mathcal{X} \to [0, 1]$ *is a continuous Borel-measurable observation function assigning a probability distribution to observation* y^x_n *given state* x_n *for all n:* $T_y(dy^x_n \in B|x_n)$

9. $T_{y^q} : \mathcal{Y}^q \times \mathcal{Q} \times \mathcal{U} \to [0, 1]$ *is a discrete observation function assigning a probability distribution to* y^q_n *given* q_n *and* u_n *for all n:* $T_{y^q}(y^q_n|q_n, u_n)$

10. $\rho : \mathcal{B}(\mathbb{R}^n) \times \mathcal{Q} \to [0, 1]$ *is an initial Borel-measurable density lying in the space of all probability measures on* $\mathcal{S}$*, such that* $\rho(dx_0 \in B, q)$

We again let $\tau_s(ds'|s,u)$ be the hybrid state transition kernel defined as the product of T_x and T_q. Similarly, the hybrid observation kernel $\tau_y(y|s,u)$ is

$$\tau_y(dy_n|s_n,u_{n-1}) = T_{y^x}(dy_n^x|x_n,u_{n-1})T_{y^q}(y_n^q|q_n,u_{n-1}). \tag{4.1}$$

Kernels T_{y^x} and T_{y^q} are more specifically defined according to

$$\begin{aligned} y_n^x &= h(x_n,u_{n-1}) + v_n, \\ y_n^q &\sim Q_{q_n,y_n^q}(u_{n-1}). \end{aligned} \tag{4.2}$$

The continuous state observation y_n^x is subject to additive noise v_n, which is independent and identically distributed with positive density $\varphi(v)$ (i.e. Gaussian), and the function h is assumed bounded and continuous, as in [JBE94]. The distribution of the discrete state observation y_n^q follows the discrete map $Q_{q,y^q}(u) : \mathcal{Q} \times \mathcal{Q} \times \mathcal{U} \to [0,1]$, so that $P[y_n^q = i \mid q_n = q,\, u_{n-1} = u] = Q_{q,i}(u)$.

The filtrations $\mathcal{G}_n$ and $\mathcal{Y}_n$ are generated by the sequences $\{s_0,\ldots,s_n,y_1,\ldots,y_{n-1}\}$ and $\{y_1,\ldots,y_n\}$, respectively. Denote $i_n = (y_1,\ldots,y_n,u_0,\ldots,u_{n-1}) \in \mathcal{Y}_n \times \mathcal{U}^n = \mathcal{I}_n$, the vector of information available at time n, with $\mathcal{U}^n$ the n-times product space of $\mathcal{U}$. The information vector i_n is available to the controller, so that policy π (defined in Chapter 2) is now a sequence of functions mapping $\mathcal{I}_n$ to $\mathcal{U}$.

The initial density ρ over $s_0 = (x_0,q_0)$ lies in the space of all probability measures on $\mathcal{S}$, $P(\mathcal{S})$ and assigns likelihoods to all possible $s_0 \in \mathcal{S}$. Finally, based on ρ, τ, ϕ, and $Q(u)$, we obtain a probability measure $\mathbb{P}^\pi$ induced by the control policy π defined over the full state space Ω, which includes s_n and y_n for all n. The PODTSHS is therefore related to a POMDP, just as a DTSHS is related to an MDP. We will elaborate upon POMDPs in Section 4.4, and exploit this relationship when examining reachability properties of a PODTSHS.

4.3.2 Reachability Analysis for a PODTSHS

Problems 2.1 - 2.3 are formulated in the same manner for a DTSHS or a PODTSHS. The cost function remains the same, and we still seek to maximize the probability of some objective over a control policy π. However, the probability measure $\mathbb{P}^\pi$ over which the probabilities (and equivalent expected value formulations) are evaluated changes. The optimal control policy also must be modified, since it is no longer a function of state s.

For instance, recall the viability problem 2.1 has cost function

$$p^N_{\text{viab}}(s_0; K) = \sup_{\pi\in\Pi} \mathbb{E}^\pi \left[\prod_{n=0}^{N} 1_K(s) \middle| s_0 \right]. \tag{4.3}$$

The state s_0 is no longer known, and so our viability probability must be a function of ρ instead. We therefore write

$$p^N_{\text{viab}}(\rho; K) = \sup_{\pi\in\Pi} \mathbb{E}^\pi \left[\prod_{n=0}^{N} 1_K(s) \middle| \rho \right]. \tag{4.4}$$

In turn, the expected value with respect to measure $\mathbb{P}^\pi$ is expanded as

$$\mathbb{E}^\pi \left[\prod_{n=0}^{N} 1_K(s) \middle| \rho \right] = \int_{\mathcal{S}}\int_{\mathcal{S}}\int_{\mathcal{Y}} \cdots \int_{\mathcal{S}}\int_{\mathcal{Y}} \prod_{n=1}^{N} 1_K(s_n)\tau_y(dy_n|s_n, u_{n-1}) \\ \times \tau_s(ds_n|s_{n-1}, u_{n-1}) 1_K(s_0)\rho(ds_0). \tag{4.5}$$

The control inputs u_n are chosen according to policy $\pi = (\mu_0, \dots, \mu_{N-1})$ with $\mu_i : \mathcal{I}_i \to \mathcal{U}$ for all i, and the optimal policy is

$$\pi^* = \arg\sup_{\pi\in\Pi} p^N_{\text{viab}}(\rho; K). \tag{4.6}$$

We therefore define a modified version of Problem 2.1.

Problem 4.1. *Consider a PODTSHS $\mathcal{H}^{po}$ (defined in Definition 4.1). Given a safe set K and time horizon N we would like to*

1. *Find the maximal probability (4.4) of remaining within K for N time steps.*

2. *Find the optimal policy π^* given by (4.6).*

In the case of perfect state information, the viability probability and optimal policy can be found via dynamic programming as described in Chapter 2, with a value function mapping $\mathcal{S}$ to probabilities in $[0,1]$. To use a dynamic program to solve Problem 4.1, the value function will be in terms of the information vector i_n, which grows over time. One approach for partially observable problems in general is a reformulation of the state i_n in terms of a sufficient statistic, and to then solve the equivalent problem using dynamic programming. This is the standard approach for POMDPs, which we describe next.

4.4 POMDP Overview

We provide an overview of POMDPs and a dynamic programming formulation for optimally controlling them. The relation between a PODTSHS and a POMDP allows us to take a similar approach when solving Problem 4.1.

A POMDP is an MDP (Definition 2.3) with the addition of an observation process. POMDPs provide a framework for analyzing a controlled discrete time system, where the controller is designed to optimize a known objective without knowledge of the current state of the system. The state evolves stochastically and is Markovian (e.g., the state at the next time step depends only on the current state and action). We first define a POMDP with finite states, actions, and observations, and an additive cost function. The theory and solution techniques for this type of POMDP provide the foundation for our extension to a PODTSHS and the solution of Problem 4.1.

Definition 4.2. *(Partially Observable Markov Decision Process $\mathcal{J}^{po}$) A POMDP is a tuple $\mathcal{J}^{po} = (\mathcal{S}, \mathcal{U}, \mathcal{Y}, T, Y, R)$ where*

1. *$\mathcal{S}$ is a finite set of states*
2. *$\mathcal{U}$ is a finite set of possible actions the agent can take*
3. *$\mathcal{Y}$ is a finite set of observations*
4. *$T : \mathcal{S} \times \mathcal{S} \times \mathcal{U} \to [0,1]$ is a state transition function assigning a probability distribution to state s_{n+1} given state s_n and action u_n for all n: $T(s_{n+1}|s_n, u_n)$*

5. $Y : \mathcal{Y} \times \mathcal{S} \times \mathcal{U} \rightarrow [0, 1]$ *is an observation function assigning a probability distribution to observation* y_n *given state* s_n *and action* u_n *for all* n*:* $Y(y_n|s_n, u_n)$

6. $R : \mathcal{S} \times \mathcal{U} \rightarrow \mathbb{R}$ *is a function assigning a reward at each time step* n*, given the current state* s_n *and action* u_n*,* $R(s_n, u_n)$

The goal is to maximize the expected sum of rewards over a (possibly infinite) time horizon N by optimally choosing a control policy π that selects inputs u_n based on available information.

$$\sup_{\pi} \mathbb{E}\left[\sum_{n=0}^{N} R(s_n, u_n)\right] \tag{4.7}$$

In Chapter 2 we discussed a Markov policy that maps the current state to a control input. However, an optimal policy in the partially observed case cannot be Markov with respect to the observations [PGT06], i.e. the controller cannot only consider the currently available observation and still make optimal decisions. One option is to consider the internal information vector i_n as defined above (the sequence of all past actions and observations) and design a policy based on the current information vector. Because the information vector grows over time, recording and storing i_n is difficult as n increases.

Instead, the optimal control at time n can be based on a belief state that summarizes all available information up to time n, as opposed to a recorded history of all past actions and observations. The belief state is a *sufficient statistic* for the set of all observations and actions $\{u_1, \ldots, u_{n-1}, y_1, \ldots, y_n\}$ because it condenses all information necessary for making optimal decisions [Ber05]. For reward function R of Definition 4.2 that is additive, the belief state is a probability density function that describes the likelihood of being in state s given all past observations and actions, $b(s_n) = P[s_n|u_1, \ldots, u_{n-1}, y_1, \ldots, y_n]$. By treating the belief state as the true state of the system, $\mathcal{J}^{po}$ can be equivalently solved as a perfect state information Markov decision process. An optimal policy π^* for the POMDP is then defined in terms of the belief state, and maps beliefs to actions: $\pi^* : \mathcal{B} \rightarrow \mathcal{U}$, with $\mathcal{B}$ the set of all possible belief states.

The optimal policy for a particular belief state can be found by maximizing a value function that describes the cumulative reward from time n to the final time N (or over $N-n+1$ time steps), presuming the system behaves optimally from time $n+1$ to N. The value function is defined recursively as

$$V_n^*(b) = \max_{u\in\mathcal{U}} \left\{ \sum_s R(s,u)b(s) + \sum_y V_{n+1}^* \left(M_{y,u}b\right) \mathbb{P}(y|u,b) \right\}, \tag{4.8}$$

with the transition operator $M_{y,u}b$ that provides the next belief state b_{n+1} given the current observation, action, and belief state

$$\left(M_{y,u}b\right)(s') = \frac{Y(y|s',b,u)\sum_{s\in\mathcal{S}} T(s'|s,u)b(s)}{\mathbb{P}(y|b,u)}, \tag{4.9}$$

and likelihood of the observation given by

$$\mathbb{P}(y|b,u) = \sum_{s\in\mathcal{S}} b(s) \sum_{s'\in\mathcal{S}} T(s'|s,u)Y(y|s',u). \tag{4.10}$$

We next adopt this formulation for verification of a PODTSHS.

4.5 Reformulation of Viability Problem

We give a detailed description of how the viability problem 4.1 with only an observation process available to the controller is reformulated. This is done in the same manner as for a POMDP, i.e. through a sufficient statistic that generates a new, perfectly observed state. The viability problem is then posed as a dynamic program over the new state. Similar results for reachability (Problem 2.2) and reach-avoid (Problem 2.3) are presented in less detail in Section 4.5.4.

The difficulty in solving Problem 4.1 is twofold. First, since the cost function is multiplicative, standard sufficient statistics are not valid (i.e. the sufficient statistic cannot be the posterior distribution of the state at time n given all available information up to time n). Second, the hybrid nature of the dynamics complicates the probability space our problem is defined on. A new sufficient statistic must be derived, and its corresponding theoretical results carefully extended.

4.5.1 Sufficient Statistic

We first formally define a sufficient statistic in relation to the multiplicative optimal control problem 4.1, which is modified from Definition 10.6 in [BS96].

Definition 4.3. *A statistic for Problem 4.1 is a sequence of Borel-measurable functions $\eta = (\eta_0, \eta_1, \dots, \eta_N)$ with $\eta_n : P(\mathcal{S}) \times \mathcal{I}_n \to \Sigma_n$ in which Σ_n is a nonempty Borel space, for all $n = 0, \dots, N$. The sequence $\eta = (\eta_0, \dots \eta_N)$ is a statistic* sufficient for control *if*

1. *There exist Borel-measurable stochastic kernels $\hat{\tau}_n(d\sigma_{n+1} \mid \sigma_n, u_n)$ on Σ_{n+1} given Σ_n, $\mathcal{U}$ such that*

$$\mathbb{P}^\pi[\eta_{n+1}(\rho, i_{n+1}) \in \underline{\Sigma}_{n+1} \mid \eta_n(\rho, i_n) = \sigma_n, u_n = u] \\ = \hat{\tau}_n(\underline{\Sigma}_{n+1} \mid \sigma_n, u) \tag{4.11}$$

   for $\mathbb{P}^\pi$-almost every $(\sigma_n, u) \in \Sigma_n \times \mathcal{U}$ (i.e. the set $\{(s_0, u_0, y_1, \dots, y_n, s_n, u_n) \in \mathcal{S}_n \times \mathcal{U}_n \times \mathcal{Y}_n \mid$ (4.11) holds when $\sigma_n = \eta_n(\rho, i_n), u_n = u\}$ has $\mathbb{P}^\pi$-measure 1), where $\underline{\Sigma}_{n+1} \in \mathcal{B}(\Sigma_{n+1})$, the Borel σ-algebra on Σ_{n+1}.

2. *There exist functions $g_n : \Sigma_n \to [0, \infty)$ such that for all $\rho \in P(\mathcal{S})$, $n = 1, \dots, N$, and $\pi \in \Pi$*

$$\mathbb{E}^\pi\left[\prod_{i=0}^{n} 1_K(s_i) \middle| \eta_n(\rho, i_n) = \sigma_n\right] = g_n(\sigma_n)$$

   for $\mathbb{P}^\pi$-almost every σ_n.

In other words, the distribution of $\sigma_n = \eta_n(\rho, i_n)$ (a specific element of the sufficient statistic at time n which we refer to as the *information state* to distinguish from the belief state of Section 4.4) must follow the Markov property, and therefore be updated recursively according to σ_{n-1} and u_{n-1}. There also must exist an equivalent cost function whose argument is σ_n, so that the cost corresponding to a specific policy can be determined solely through the distribution of the information state. Problem 4.1 can then be redefined according to the information state σ, which is defined according to the sufficient statistic η.

We now propose a sufficient statistic for Problem 4.1, and demonstrate that it obeys Definition 4.3.

We claim that under the measure $\mathbb{P}^\pi$ we can define a sufficient statistic for Problem 4.1.

$$\eta_n(\rho, i_n) = \mathbb{E}^\pi\left[1_{dx}(x_n) 1_q(q_n) \prod_{i=0}^{n-1} 1_K(s_i) \middle| \rho, i_n \right] \tag{4.12}$$

The statistic $(\eta_0, \eta_1, \ldots, \eta_N)$ given by (4.12) for all n generates a sequence of probability densities on the state s_n, combined with the probability that all previous states are in K, conditioned on the information vector i_n, with dependence on ρ implicitly defined in the measure $\mathbb{P}^\pi$.

$$\eta_n(\rho, i_n) = \mathbb{P}^\pi[x_n \in dx, q_n = q, s_0, \ldots, s_{n-1} \in K | i_n] \tag{4.13}$$

The information state σ_n is therefore a modification of the posterior distribution, representing a conditional density of the current state joined with the probability that all previous states are in K, given a specific i_n. In the following, we show that (4.12) satisfies conditions (1) and (2) of Definition 4.3, and is therefore a sufficient statistic, by examining the information state $\sigma_n = \eta_n(\rho, i_n)$ directly. Although η_n is a function of ρ and i_n, σ_n is a function of the state s_n, and is implicitly dependent on ρ and i_n. We first show that σ_n can be defined recursively via a bounded linear operator $\Phi_{y,u}$, and therefore satisfies (1) of Definition 4.3.

Lemma 4.1. *There exists a bounded linear operator $\Phi : L^1(\mathcal{S}) \to L^1(\mathcal{S})$ such that σ is defined recursively as*

$$\begin{aligned} \sigma_0 &= \rho \\ \sigma_n &= \Phi_{y_n, u_{n-1}} \sigma_{n-1} \end{aligned} \tag{4.14}$$

with $\Phi_{y,u}\sigma$ given by

$$\left(\Phi_{y,u}\sigma\right)(s') = \frac{1}{\mathbb{P}(y|\sigma, u)} \tau_y(y|s', u) \int_K \tau_s(s'|s, u)\sigma(s)\, ds. \tag{4.15}$$

Note we have incorporated the factor $\mathbb{P}(y|\sigma, u)$ equal to $\int_{\mathcal{S}} \int_{\mathcal{S}} \tau_y(y|s', u)\tau_s(ds'|s, u)\sigma(ds)$ that ensures σ integrates to one over $\mathcal{S}^n$ (i.e. over all $\mathcal{S}$ and either $\prod_{i=0}^n 1_K(s_i) = 0$ or $\prod_{i=0}^n 1_K(s_i) = 1$). In addition, $\sigma_n \in L^1(S)$ for all n since $\sigma_0 = \rho \in L^1(\mathcal{S})$ and Φ maps $L^1(\mathcal{S})$ into $L^1(\mathcal{S})$.

Proof. We first show that Φ is a bounded linear operator mapping $L^1(\mathcal{S})$ to itself. We then show that σ_n can be defined recursively using Φ. We drop the normalizing constant $\mathbb{P}(y|\sigma, u)$ for convenience, because it does not affect subsequent derivations.

Linearity follows obviously from the properties of integrals. For any function $\nu \in L^1(\mathcal{S})$, $u \in \mathcal{U}$, $y \in \mathcal{Y}$,

$$\begin{aligned}
\|\Phi_{y,u}\nu\|_{L^1(\mathcal{S})} &= \int_{\mathcal{S}} \left| \tau_y(y|s', u) \int_K \tau_s(ds'|s, u)\nu(ds) \right| && (4.16)\\
&\leq \int_K \int_{\mathcal{S}} T_{y^q}(y^q|q', u)\phi(y^x - h(x', u))\tau_s(ds'|s, u)|\nu(ds)| && (4.17)\\
&\leq \phi_y^* \int_K \left[\int_{\mathcal{S}} \tau_s(ds'|s, u) \right] |\nu(ds)| && (4.18)\\
&\leq \phi_y^* \|\nu\|_{L^1(\mathcal{S})} && (4.19)
\end{aligned}$$

with ϕ_y^* the maximum value of density ϕ (which may exceed one). Hence for any $\nu \in L^1(\mathcal{S})$, Φ is a bounded linear operator, with $\Phi_{y,u}\nu \in L^1(\mathcal{S})$.

Induction shows that $\sigma_n = \Phi_{y,u}\sigma_{n-1}$. Given $\sigma_0 = \rho$,

$$\begin{aligned}
\left(\Phi_{y,u}\sigma\right)(x, q) &= \tau_y(y|x, q, u) \int_K \tau_s(x, q|s_0, u)\rho(ds_0)\\
&= \mathbb{E}^\pi \left[1_q(q_1)1_x(x_1)1_K(s_0) \middle| \rho, i_1 \right]\\
&= \sigma_1(x_1, q_1)
\end{aligned}$$

Given $\sigma_l = \Phi_{y,u}\sigma_{l-1} \, \forall \, l = 2, \ldots, n$,

$$\begin{aligned}
\left(\Phi_{y,u}\sigma_n\right)(x, q) &= \tau_y(y|x, q, u) \int_K \tau_s(x, q|s, u)\sigma_n(ds_n)\\
&= \tau_y(y|x, q) \int_K \cdots \int_K \tau_s(x, q|s_{n-1}, u)\\
&\qquad \times \prod_{i=1}^n \tau_y(y_i|s_i, u_{i-1})\tau_s(ds_i|s_{i-1}, u_{i-1})\rho(ds_0)\\
&= \mathbb{E}^\pi \left[1_x(x_{n+1})1_q(q_{n+1}) \prod_{i=0}^n 1_K(s_i) \middle| \rho, i_n \right]
\end{aligned}$$

■

The stochastic kernel $\hat{\tau}_n$ for the distribution of σ_{n+1} given σ_n and u_n can be written in terms of the new measure $\mathbb{P}^\pi$:

$$\begin{aligned}\hat{\tau}_n(\underline{\Sigma}_{n+1}|\sigma_n, u_n) &= \int_{\mathcal{S}}\int_{\mathcal{S}}\int_{\underline{Y}} \tau_y(dy|s_{n+1}, u_n)\tau_s(ds_{n+1}|s_n, u_n)\sigma_n(ds_n) \\ &= \mathbb{P}(\underline{Y}|\sigma_n, u_n)\end{aligned} \tag{4.20}$$

with $\underline{Y} = \{y : \Phi_{y,u_n}\sigma_n \in \underline{\Sigma}_{n+1}\}$. Note the similarity between (4.20) and (4.10). The stochastic transition kernel for σ is directly related to the likelihood of observing some $y \in \mathcal{Y}$.

Next, we rewrite the cost function (4.4) in terms of the information state σ, for part (2) of Definition 4.3. Since the indicator function $1_K(s)$ is in the space $L^\infty(\mathcal{S})$, the inner product of σ and 1_K is a well defined bounded linear operator, given by

$$\langle \sigma, 1_K \rangle = \sum_{q\in\mathcal{Q}} \int_{\mathbb{R}^n} \sigma(x, q) 1_K(x, q)\, dx$$

The functions g_n in Definition 4.3 can be defined as

$$\begin{aligned}g_n(\sigma_n) =& \mathbb{E}^\pi\left[\prod_{i=0}^{n} 1_K(s_i)\middle|\, \eta_n(\rho, i_n) = \sigma_n\right] \\ =& \mathbb{E}^\pi\left[1_K(s_n)\prod_{i=0}^{n-1} 1_K(s_i)\middle|\, \rho, i_n\right] \\ =& \langle \sigma_n, 1_K\rangle\end{aligned} \tag{4.21}$$

Hence the statistic given by (4.12) satisfies parts (1) and (2) of Definition 4.3, and is a sufficient statistic for Problem 4.1.

4.5.2 Equivalence to Perfect State Information Problem

We can rewrite the viability probability in terms of σ and function $g_n(\sigma)$. First, we define

$$\tilde{p}^N_{\text{viab}}(\rho; K) = \mathbb{E}^\pi\left[g_N(\sigma_N)\right] \tag{4.22}$$

as the equivalent cost function for the viability problem in terms of σ, using $\tilde{\pi}$ to denote a policy in terms of σ (whereas π denotes a policy for the partially observed case in terms of i_n). Note that for a fixed vector of control inputs (i.e. open-loop), $\mathbf{u} = [u_0, \ldots, u_{N-1}]$,

$$\begin{aligned} p^N_{\text{viab}}(\rho; K) &= \mathbb{E}^\pi \left[\prod_{i=0}^{N} 1_K(s_i) \middle| \rho \right] \\ &= \mathbb{E}^\pi \left[\mathbb{E}^\pi \left[\prod_{i=0}^{N} 1_K(s_i) \middle| \rho, i_N \right] \right] \\ &= \mathbb{E}^\pi \left[g_N(\sigma_N) \right] \\ &= \tilde{p}^N_{\text{viab}}(\rho; K). \end{aligned}$$

Lemma 4.2. *A recursion for $\tilde{p}^N_{\text{viab}}(\rho; K)$ is given by*

$$\begin{aligned} V^{\tilde{\pi}}_N(\sigma) &= g_N(\sigma) \\ V^{\tilde{\pi}}_n(\sigma) &= \mathbb{E}^{\tilde{\pi}} \left[V^{\tilde{\pi}}_{n+1}(\Phi_{y,\tilde{\mu}_n(\sigma)}\sigma) \right] \end{aligned} \tag{4.23}$$

where

$$\mathbb{E}^{\tilde{\pi}} \left[V^{\tilde{\pi}}_{n+1}(\Phi_{y,\tilde{\mu}_n(\sigma)}) \right] = \int_\Sigma V_{n+1}(\sigma')\hat{\tau}(\sigma'|\sigma, \tilde{\mu}_n(\sigma))\, d\sigma' \tag{4.24}$$

$$= \int_{\mathcal{S}} \int_{\mathcal{S}} \int_{\mathcal{Y}} V_{n+1}(\Phi_{y,\tilde{\mu}_n(\sigma)}\sigma)\tau_y(dy|s', \tilde{\mu}_n(\sigma)) \times \tau_s(ds'|s, \tilde{\mu}_n(\sigma))\sigma(ds) \tag{4.25}$$

so that $V^{\tilde{\pi}}_0(\rho) = \tilde{p}^N_{\text{viab}}(\rho; K)$ with $\tilde{p}^N_{\text{viab}}(\rho; K)$ computed using policy $\tilde{\pi}$ rather than the supremum of all policies.

Proof. First, we redefine $V^{\tilde{\pi}}_n(\sigma)$ as

$$V^{\tilde{\pi}}_n(\sigma) = \int_{\mathcal{S}} \int_{\mathcal{Y}} \cdots \int_{\mathcal{S}} \int_{\mathcal{Y}} \left[\prod_{i=n+1}^{N} 1_K(s_i)\tau_y(dy_i|s_i, \tilde{\mu}_{i-1}(\sigma_{i-1})) \times \tau_s(ds_i|s_{i-1}, \tilde{\mu}_{i-1}(\sigma_{i-1})) \right] 1_K(s_n)\sigma(ds_n)$$

for all $n = 0, \ldots, N-1$, and

$$V^{\tilde{\pi}}_N(\sigma) = \int_{\mathcal{S}} 1_K(s)\sigma(ds),$$

and then show that $V_n^{\tilde{\pi}}$ can be expressed recursively according to $V_{n+1}^{\tilde{\pi}}$. Clearly, $V_0^{\tilde{\pi}}(\rho)$ is equal to (4.5). The proof proceeds by induction. At time $N-1$,

$$\begin{aligned} V_{N-1}^{\tilde{\pi}}(\sigma) &= \int_{\mathcal{S}}\int_{\mathcal{S}}\int_{\mathcal{Y}} 1_K(s_N)\tau_y(dy_N|s_N,\tilde{\mu}_{N-1}(\sigma)) \\ &\qquad\qquad \times \tau_s(ds_N|s_{N-1},\tilde{\mu}_{N-1}(\sigma))1_K(s_{N-1})\sigma(ds_{N-1}) \\ &= \int_{\mathcal{Y}} V_N^{\tilde{\pi}}(\Phi_{y,\tilde{\mu}_{N-1}(\sigma)}\sigma)\hat{\tau}(\Phi_{y,\tilde{\mu}_{N-1}(\sigma)}\sigma|\sigma,\tilde{\mu}_{N-1}(\sigma)) \\ &= \mathbb{E}^{\tilde{\pi}}\left[V_N^{\tilde{\pi}}(\Phi_{y,\tilde{\mu}_{N-1}(\sigma)})\right]. \end{aligned}$$

At time n,

$$\begin{aligned} V_n^{\tilde{\pi}}(\sigma) = (\sigma) &\int_{\mathcal{S}}\int_{\mathcal{Y}}\cdots\int_{\mathcal{S}}\int_{\mathcal{Y}}\left[\prod_{i=n+2}^{N} 1_K(s_i)\tau_y(dy_i|s_i,\tilde{\mu}_{i-1}(\sigma_{i-1}))\right. \\ &\qquad \left.\times\ \tau_s(ds_i|s_{i-1},\tilde{\mu}_{i-1}(\sigma_{i-1}))\right] 1_K(s_{n+1})\tau_y(dy_{n+1}|s_{n+1},\tilde{\mu}_n(\sigma)) \\ &\qquad \times\ \tau_s(ds_{n+1}|s_n,\tilde{\mu}_n(\sigma))1_K(s_n)\sigma(ds_n) \\ = &\int_{\mathcal{Y}} V_{n+1}^{\tilde{\pi}}(\Phi_{y,\tilde{\mu}_n(\sigma)})\hat{\tau}(\Phi_{y,\tilde{\mu}_n(\sigma)}) \\ = &\ \mathbb{E}^{\tilde{\pi}}\left[V_{n+1}^{\tilde{\pi}}(\Phi_{y,\tilde{\mu}_n(\sigma)})\right]. \end{aligned}$$

■

Next, we provide two theorems: 1) a dynamic programming algorithm to find the optimal solution $\tilde{p}_{\text{viab}}^N(\rho;K)$, and the optimal policy $\tilde{\pi}^* = \arg\sup_{\tilde{\pi}} \tilde{p}_{\text{viab}}^N(\rho;K)$ as a function of the information state σ, and 2) proof that this optimal policy has the same value as the optimal policy for the partially observed case. The proofs of Theorems 4.1 and 4.2 are provided in Appendix A.

Theorem 4.1. *Using the recursion (4.23), the dynamic programming equations*

$$\begin{aligned} V_N^*(\sigma) &= \langle\sigma, 1_K\rangle \\ V_n^*(\sigma) &= \sup_{u\in\mathcal{U}} \mathbb{E}^{\tilde{\pi}}\left[V_{n+1}^{\tilde{\pi}}(\Phi_{y,u}\sigma)\right] \end{aligned} \tag{4.26}$$

produce $V_0^*(\rho) = \tilde{p}_{\text{viab}}^N(\rho;K)$, where $V_n : \Sigma_n \to [0,\infty)$. For σ normalized, we have $V_n : \Sigma_n \to [0,1]$. Furthermore, setting

$$\tilde{\mu}_n^*(\sigma) = \arg\sup_{u\in\mathcal{U}} \mathbb{E}^{\tilde{\pi}}\left[V_{n+1}^*(\Phi_{y,u}\sigma)\right] \tag{4.27}$$

for all $n = 0, \dots, N-1$ *gives the optimal policy* $\tilde{\pi}^* = (\tilde{\mu}_0^*, \tilde{\mu}_1^*, \dots, \tilde{\mu}_{N-1}^*)$*, where* $\tilde{\mu}_n : \Sigma_n \to \mathcal{U}$*.*

Theorem 4.2. *If* $u_n^* = \tilde{\mu}_n^*(\sigma_n)$ *is optimal as defined in Theorem 4.1, then* u_n^* *is also optimal for the partially observable Problem 4.1, and can be written as* $u_n^* = \mu_n^*(i_n) = \tilde{\mu}_n^*(\eta_n(\rho, i_n)) = \tilde{\mu}_n^*(\sigma_n)$*. For* $\pi^* = (\mu_0^*, \mu_1^*, \dots, \mu_{N-1}^*)$*,* $\tilde{p}_{\text{viab}}^N(\rho; K) = p_{\text{viab}}^N(\rho; K)$*.*

These results guarantee that we can solve Problem 4.1 as a fully observed problem for each n in terms of the new state σ, and generate a policy $\tilde{\pi}$ in which the optimal action at each time step n is a function only of the information state at time n. The policy is therefore a continuous mapping from information states to the control space. Calculating the optimal policy $\tilde{\pi}^*$ and optimal value $\tilde{p}_{\text{viab}}^N(\rho; K)$ gives us the optimal policy π^* and optimal value $p_{\text{viab}}^N(\rho; K)$.

4.5.3 Sufficient Statistic with Change of Measure

We also derive a sufficient statistic for Problem 4.1 that incorporates a change of measure, which renders each observation process $\{y_n^x\}$ and $\{y_n^q\}$ independent and identically distributed (i.i.d.). The change of measure simplifies the transition function $\hat{\tau}$ for the information state (4.20), and subsequently the value function (4.23).

The ability to change probability measures stems from the Radon-Nikodym Theorem.

Definition 4.4. *The Radon-Nikodym Theorem (see [SS05]) states that given two* σ*-finite measures* ν *and* μ *on a measurable space* $(\Omega, \mathcal{M})$*, if* μ *and* ν *are absolutely continuous, then there exists a* μ*-integrable function* f *on* Ω *such that*

$$\nu(E) = \int_E f(\omega)\, d\mu(\omega)$$

The function f *is referred to as the Radon-Nikodym derivative, and is written as* $\frac{d\nu}{d\mu}$*.*

Essentially, for two probability measures ν and μ, defined on the same space $(\Omega, \mathcal{M})$ and that satisfy $\nu(E) = 0$ whenever $\mu(E) = 0$ for all $E \in \mathcal{M}$, we know that

$$\mathbb{E}^{\nu}[h(\omega)] = \mathbb{E}^{\mu}[f(\omega)h(\omega)]$$

for any $\mathcal{M}$-measurable function h. We can define a change of measure $\mathbb{P}^\dagger$ from the known measure $\mathbb{P}^\pi$ on the space Ω, with $\mathcal{M}$ the Borel σ-algebra on Ω, so long as the continuous observation process is nowhere zero, and the discrete observation is nowhere zero on $\mathcal{Q} \times \mathcal{Q} \times \mathcal{U}$ (which would occur if certain discrete states were perfectly observable). Following [JBE94] and [Ell93], we define the Radon-Nikodym derivative Λ_n as

$$\left.\frac{d\mathbb{P}^\pi}{d\mathbb{P}^\dagger}\right|_{\mathcal{G}_n} = \Lambda_n \tag{4.28}$$

where

$$\Lambda_n = \prod_{l=1}^{n} \frac{\varphi(y_l^x - h(x_l, u_{l-1}))Q_{q_l, y_l^q}(u_{l-1})}{\varphi(y_l^x)\frac{1}{N_q}}$$

However, in contrast to [JBE94], we must contend with two separate observation processes, one continuous and one discrete. Note that in (4.28) we restrict the derivative to the filtration $\mathcal{G}_n$ rather than the full state space Ω, which enables updates to the derivative as the hybrid process evolves.

Intuitively, the change of measure introduces a proposal distribution that can be treated as a design parameter, as used in importance sampling and particle filtering (see e.g. [MSAC02]) when the true distribution is difficult to sample from. This is particularly relevant when solving Problem 4.1 numerically, as sampling-based solutions are likely required. Further discussion of this point, and the role of the change of measure, is presented in Section 4.6.

Lemma 4.3. *Under $\mathbb{P}^\dagger$, the processes $\{y_n^x\}$ are independent and identically distributed (i.i.d.) with density ϕ, and the processes $\{y_n^q\}$ are i.i.d. with uniform density $\frac{1}{N_q}$.*

Proof. The proof follows that of [Ell93].

$$\begin{aligned}\mathbb{P}^\dagger\left[y_n^x \in A,\, y_n^q = q \mid \mathcal{G}_n\right] &= \mathbb{E}^\dagger\left[1_A(y_n^x)1_{\{q\}}(y_n^q) \mid \mathcal{G}_n\right] \\ &= \frac{\mathbb{E}^\pi\left[1_A(y_n^x)1_{\{q\}}(y_n^q)\Lambda_n^{-1} \mid \mathcal{G}_n\right]}{\mathbb{E}^\pi\left[\Lambda_n^{-1} \mid \mathcal{G}_n\right]}\end{aligned}$$

Pulling Λ_{n-1}^{-1} outside the expected value from both the numerator and denominator and canceling, since Λ_{n-1} is $\mathcal{G}_n$-measurable, the numerator reduces to

$$\begin{aligned}&\int_A \frac{\varphi(y)\frac{1}{N_q}}{\varphi(y-h(x_n,u_{n-1}))Q_{q_n,q}(u_{n-1})}\\ &\qquad\qquad \times \mathbb{P}^{\pi}\left[y_n^x = y, y_n^q = q | s_n, u_{n-1}\right] dy\\ &= \frac{1}{N_q}\int_A \phi(y)dy = \mathbb{P}^{\dagger}\left[y_n^q = q\right]\mathbb{P}^{\dagger}\left[y_n^x \in A\right]\end{aligned}$$

and the denominator becomes

$$\begin{aligned}&\sum_{q=1}^{N_q}\int_{\mathbb{R}^n} \frac{\varphi(y)\frac{1}{N_q}}{\varphi(y-h(x_n,u_{n-1}))Q_{q_n,q}(u_{n-1})}\\ &\qquad\qquad \times \mathbb{P}^{\pi}\left[y_n^x = y, y_n^q = q | s_n, u_{n-1}\right] dy\\ &= \sum_{q=1}^{N_q}\frac{1}{N_q}\int_{\mathbb{R}^n} \varphi(y)dy = 1\end{aligned}$$

Hence,

$$\mathbb{P}^{\dagger}\left[y_n^x \in A,\, y_n^q = q \mid \mathcal{G}_n\right] = \mathbb{P}^{\dagger}\left[y_n^q = q\right]\mathbb{P}^{\dagger}\left[y_n^x \in A\right]$$

■

Under the change of measure, we define a new sufficient statistic $\overline{\eta} = (\overline{\eta}_0, \dots, \overline{\eta}_N)$ such that

$$\overline{\eta}_n(\rho, i_n) = \mathbb{E}^{\dagger}\left[1_q(q_n)1_x(x_n)\prod_{i=0}^{n-1}1_K(s_i)\Lambda_n\middle| i_n\right] \tag{4.29}$$

and $\overline{\eta}_n(\rho, i_n) = \overline{\sigma}_n(s)$. The linear operator $\overline{\Phi}_{y,u}$ adapted from (4.15) is

$$\left(\overline{\Phi}_{y,u}\overline{\sigma}\right)(x', q') = N_q Q_{q',y^q}(u)\frac{\varphi(y^x - h(x',u))}{\varphi(y^x)}\int_K \tau_s(x', q'|x)\overline{\sigma}(ds)\,dx \tag{4.30}$$

Operator $\overline{\Phi}$ shares the same properties as Φ (bounded linear operator) which can be established in the same manner as for Lemma 4.1.

To show $\overline{\eta}$ is a sufficient statistic, and hence satisfies the conditions of Definition 4.3, we define functions $\hat{\overline{\tau}}$ and $\overline{g}$.

$$\hat{\overline{\tau}}_n(\underline{\Sigma}_{n+1}|\overline{\sigma}_n, u_n) = \sum_{y^q\in\underline{Y}^q}\int_{\underline{Y}^x}\frac{1}{N_q}\varphi(y^x)\,dy^x \tag{4.31}$$

with $\underline{Y}^q \times \underline{Y}^x = \{(y^q, y^x) : \overline{\Phi}_{y,u}\overline{\sigma} \in \underline{\Sigma}_{n+1}\}$.

$$\begin{aligned}\overline{g}_n(\overline{\sigma}_n) =&\mathbb{E}^\pi\left[\prod_{i=0}^{n} 1_K(s_i)\middle|\overline{\eta}_n(\rho, i_k) = \overline{\sigma}_n\right]\\ =&\mathbb{E}^\dagger\left[1_K(s_n)\prod_{i=0}^{n-1} 1_K(s_i)\Lambda_n\middle| i_n\right]\\ =&\langle\overline{\sigma}_n, 1_K\rangle\end{aligned} \tag{4.32}$$

The viability probability (4.4) is again rewritten in terms of $\overline{\sigma}$.

$$\overline{p}^N_{\text{viab}}(\rho; K) = \mathbb{E}^\dagger\left[\langle\overline{\sigma}_N, 1_K\rangle\right] \tag{4.33}$$

The dynamic programming recursion

$$\begin{aligned}\overline{V}^*_N(\overline{\sigma}) &= \langle\overline{\sigma}, 1_K\rangle\\ \overline{V}^*_n(\overline{\sigma}) &= \sup_{u\in\mathcal{U}}\mathbb{E}^\dagger\left[\overline{V}^*_{n+1}(\overline{\Phi}_{y,u}\overline{\sigma})\right]\\ &= \sup_{u\in\mathcal{U}}\sum_{y^q\in\mathcal{Y}^q}\int_{\mathbb{R}^n}\overline{V}^*_{n+1}(\overline{\Phi}_{y,u}\overline{\sigma})\frac{1}{N_q}\varphi(dy^x)\end{aligned} \tag{4.34}$$

produces $\overline{V}^*_0(\rho) = \overline{p}^N_{\text{viab}}(\rho; K)$. The equivalence of $\overline{p}^N_{\text{viab}}(\rho; K)$ and $\tilde{p}^N_{\text{viab}}(\rho; K)$ follows directly from the Radon-Nikodym Theorem.

4.5.4 Sufficient Statistic for Reachability and Reach-Avoid Problems

Sufficient statistics for the optimal control formulations of the reachability and reach-avoid problems can be derived in the same manner as for viability. We do not provide full derivations as for the viability sufficient statistic, but give the information state and its transition function, and the dynamic program to solve the partially observable versions of Problems 2.2 and 2.3.

We give the reach-avoid sufficient statistic first, because a reachability objective follows directly from the reach-avoid or viability formulation. Recall the optimal control formulation for the reach-avoid problem has cost function

$$p_{\mathrm{ra}}^{N}(\rho; K, T) = \sup_{\pi \in \Pi} \mathbb{E}^{\pi}\left[\sum_{k=0}^{N}\left(\prod_{n=0}^{k-1} 1_{K\setminus T}(\mathbf{s}_n)\right) 1_T(\mathbf{s}_k) \middle| \rho\right] \tag{4.35}$$

with initial state s_0 replaced by distribution ρ to reflect the partial observability of the state. The sufficient statistic, which we denote $\eta^{\mathrm{ra}} = (\eta_0^{\mathrm{ra}}, \ldots, \eta_N^{\mathrm{ra}})$, provides functions

$$\eta_n^{\mathrm{ra}}(\rho, i_n) = \mathbb{E}^{\pi}\left[1_x(x_n) 1_q(q_n) \prod_{i=0}^{n-1} 1_{K\setminus T}(s_i) \middle| \rho, i_n\right] \tag{4.36}$$

such that $\eta_n^{\mathrm{ra}}(\rho, i_n) = \sigma_n^{\mathrm{ra}}(s)$. The information state can be defined recursively according to

$$\begin{aligned} \sigma_0^{\mathrm{ra}} &= \rho \\ \sigma_n^{\mathrm{ra}} &= \Phi_{y,u}^{\mathrm{ra}} \sigma_{n-1} \\ \left(\Phi_{y,u}^{\mathrm{ra}} \sigma^{\mathrm{ra}}\right)(s') &= \frac{1}{\mathbb{P}(y|\sigma^{\mathrm{ra}}, u)} \tau_y(y|s', u) \int_{K\setminus T} \tau_s(s'|s, u) \sigma(ds) \end{aligned} \tag{4.37}$$

with normalizing constant $\mathbb{P}(y|\sigma^{\mathrm{ra}}, u)$ equal to $\int_{\mathcal{S}} \int_{\mathcal{S}} \tau_y(y|s', u) \tau_s(ds'|s, u) \sigma(ds)$. Compared to (4.12), the sufficient statistic for the reach-avoid problem produces a probability density over $\mathcal{S}$ coupled with the probability that all previous states have remained within $K\setminus T$ rather than K alone. The dynamic program changes to reflect the sum-multiplicative cost function (4.35).

$$\begin{aligned} V_{N,\mathrm{ra}}^{*}(\sigma^{\mathrm{ra}}) &= \langle 1_T, \sigma^{\mathrm{ra}} \rangle \\ V_{n,\mathrm{ra}}^{*}(\sigma^{\mathrm{ra}}) &= \sup_{u \in \mathcal{U}} \mathbb{E}^{\tilde{\pi}}\left[1_T(\mathbf{s}) + 1_{K\setminus T}(\mathbf{s}) V_{n+1,\mathrm{ra}}^{*}(\Phi_{y,u}^{\mathrm{ra}} \sigma^{\mathrm{ra}})\right] \\ &= \sup_{u \in \mathcal{U}} \int_T \sigma^{\mathrm{ra}}(ds) + \int_{\mathcal{S}\setminus T} \int_{\mathcal{S}} \int_{\mathcal{Y}} V_{n+1,\mathrm{ra}}^{*}(\Phi_{y,u}^{\mathrm{ra}} \sigma) \tau_y(dy|s', u) \tau_s(ds'|s, u) \sigma^{\mathrm{ra}}(ds) \end{aligned} \tag{4.38}$$

It is straightforward to show that (4.35) equals $V_{0,\mathrm{ra}}^{*}(\rho)$. We can also verify the value function recursion similarly to the proof of Lemma 4.2. If we redefine $V^{\tilde{\pi}}_{n,\mathrm{ra}}(\sigma^{\mathrm{ra}})$ as

$$
\begin{aligned}
V^{\tilde{\pi}}_{N,\mathrm{ra}}(\sigma^{\mathrm{ra}}) &= \int_{\mathcal{S}} 1_T(s)\sigma^{\mathrm{ra}}(ds) \\
V^{\tilde{\pi}}_{n,\mathrm{ra}}(\sigma^{\mathrm{ra}}) &= \int_{\mathcal{S}} 1_T(s_n)\sigma^{\mathrm{ra}}(ds_n) + \int_{\mathcal{S}}\int_{\mathcal{Y}} \cdots \int_{\mathcal{S}}\int_{\mathcal{Y}} \left[\sum_{j=n+1}^{N} \left(\prod_{i=n+1}^{j-1} 1_{K\setminus T}(s_i) \right) 1_T(s_j) \right. \\
&\qquad \left. \times \prod_{j=n+1}^{N} \tau_y(dy_j|s_j, \tilde{\mu}_{j-1}(\sigma_{j-1}))\tau_s(ds_j|s_{j-1}, \tilde{\mu}_{j-1}(\sigma_{j-1})) \right] 1_{K\setminus T}(s_n)\sigma^{\mathrm{ra}}(ds_n)
\end{aligned}
\tag{4.39}
$$

for $n = 0, \ldots, N-1$, an induction argument shows the validity of the recursive expression for $V^{\tilde{\pi}}_{n,\mathrm{ra}}$ in terms of $V^{\tilde{\pi}}_{n+1,\mathrm{ra}}$.

The reachability problem with partial observations, $p^N_{\mathrm{reach}}(\rho; T)$, can be solved using the reach-avoid sufficient statistic (4.36) and dynamic program (4.38) by setting $K = \mathcal{S}$, or by solving an equivalent viability problem as explained in Chapter 2.

4.5.5 Relationship to Additive Cost Formulation

The sufficient statistic (4.12) modifies the posterior distribution to include the probability that all previous states are in the set K. The sufficient statistic (4.12) derived without the change of measure is identical (aside from a normalizing constant) to the sufficient statistic for the additive cost function formulation in [DAT13] (see equation (14)). In [DAT13], by extending the state to include a binary variable that represents whether or not the system has remained within K up to the previous time step, the posterior distribution is also the distribution of the current state s_n, coupled with the distribution of all previous states being in K. The transition kernel for the modified state in equation (5) of [DAT13] incorporates an indicator function that signals whether the state remained within the safe set at the previous time step. The prediction and update steps for a Bayesian filter (see equations (11) - (13) of [DAT13]) are used to express the sufficient statistic (14), which, without considering the change of measure and normalization, is the same as (4.12).

The terminal payoff (15) of [DAT13] expresses the probability that the final state is in set K, and that all previous states are in set K, given the probability distribution of all previous states in K as well as the current distribution of the final state. If the terminal payoff was written in terms of the original state, it would be identical to (4.21) for $n = N$. Prop. 3 in [DAT13] describes the solution of the terminal payoff, and iteratively evaluates the expected value as in Theorem 4.1 (although integrating the expected value over $\hat{\tau}_n(\sigma_{n+1} \mid \sigma_n, u_n)$ does not reduce to (4.34) as ours does with the change of measure). Thus, formulating the cost function as either multiplicative or additive ultimately does not alter the end result.

4.6 Case Studies and Numerical Issues

We provide two examples in order to a) demonstrate the use of the viability sufficient statistic with the change of measure in formulating various types of imperfect information reachability problems, b) elucidate the difficulty in solving these problems exactly, even when the problem seems relatively simple, and c) initiate a discussion of possible approximation strategies, given that the problems cannot be solved in their current form. Since solving (4.26) requires iterating over all functions $\sigma \in L^1(\mathcal{S})$, an infinite space, we can only hope to use (4.26) as a practical solution method for special cases in which σ can be defined over a finite subspace of L^1.

4.6.1 Temperature Regulation

A stochastic version of the benchmark temperature regulation problem with perfect state information is presented in [APLS08]. We consider the case of one heater, which can either be turned off, or turned on to heat one of M rooms. The average temperature of room i at time n is given by the continuous variable $x_i(n)$, and the discrete state $q(n) = i$ indicates room $i \in \{1, \ldots, M\}$ is heated at time n, and $q(n) = 0$ denotes the heater is off. The stochastic difference equation governing the average temperature for room i is given by

$$x_i(n+1) = (1-b_i)x_i(n) + \sum_{i \neq j} a_{i,j}(x_j(n) - x_i(n)) + c_i h_i + b_i x_a + v_i(n). \quad (4.40)$$

The constant x_a represents the ambient temperature, non-negative constants b_i and $a_{i,j}$ represent the average rates of heat loss to the ambient temperature and to other rooms $j \neq i$, respectively, and constant c_i represents the rate of heat being supplied by the heater in room i. The disturbances $v_i(n)$ are i.i.d. normally distributed random variables with mean zero and variance v^2, representing uncertainty in the model resulting from the linearized discretization of the dynamics and other uncontrollable factors. The variable h_i is a function of $q(n)$, with $h_i = 1$ for $q(n) = i$ and $h_i = 0$ otherwise. The control input is given by $u(n) \in \mathcal{U}$ with $\mathcal{U} = \{0, 1, \ldots, M\}$, but the chosen control is not always implemented with probability 1. Instead, $q(n)$ is updated probabilistically, dependent on $u(n-1)$ and $q(n-1)$, with transition function $T_q(q(n+1) \mid q(n), u(n))$. So while function $\tilde{\mu}_n(\sigma_n)$ deterministically returns a single control input, control input $u(n) = \tilde{\mu}_n(\sigma_n)$ may not always be implemented.

The average temperature in each room is unknown, and only a noisy measurement $y^x(n)$ of each room's temperature is available to the controller at each time step n. The controller does, however, know which room is heated at time n (i.e. $q(n)$ is perfectly observed). The observation $y(n) = (y^x(n), y^q(n))$ with $y^x(n) = [y_1^x(n) \ldots y_n^x(n)]^T$ is given by

$$\begin{aligned} y_i^x(n) &= x_i(n) + w_i(n) \\ y^q(n) &= q(n) \end{aligned}$$

with $w_i(n)$ i.i.d. normally distributed with mean zero and variance w^2 (so that the distribution $\phi(w)$ is Gaussian). The transition matrix $Q(u)$ is the identity matrix for all u, so that $Q_{q,y^q}(u) = 1_q(y^q)$. Because the discrete state is perfectly observed, we do not keep track of a discrete observation, and it is not included in the sufficient statistic.

The temperature of each room should stay between 17.5 and 22 degrees Celsius at all times, producing the safe region $K = [17.5, 22] \times \ldots \times [17.5, 22]$, which does not depend on the discrete state $q(n)$ (so $1_K(s) = 1_K(x)$). To find the maximum probability that each room's temperature remains within K given that the controller only has access to the observations $y(n)$, we reformulate the problem in terms of the information state σ_n. The dynamic programming equations (4.26) are given by

$$\begin{aligned} V_N^*(\sigma) &= \sum_{q=0}^{M} \int_{\mathbb{R}^n} 1_K(x)\sigma(x,q)\,dx \\ V_n^*(\sigma) &= \max_{u\in\mathcal{U}} \int_{\mathbb{R}^n} V_{n+1}^*(\Phi_{y,u}\sigma)\phi(y^x)\,dy^x \end{aligned} . \tag{4.41}$$

With $x_i(0) \sim \mathcal{N}(\mu_i, s^2)$ for each $i = 1, \dots, M$ and $q(0) = 0$, the initial information state is

$$\sigma_0(x,q) = 1_0(q)\prod_{i=1}^{M} \rho_i(x_i)$$

for $\rho_i(x)$ Gaussian with mean μ_i and variance s^2.

However, even for the trivial case of $M = 1$ (e.g. a one room system), updating σ_n becomes complicated very quickly. Using Lemma 4.1, we obtain

$$\sigma_1(x,q) = \frac{\phi(y_1^x - x)}{\phi(y_1^x)} T_q(q|q(0) = 0, u(0)) \\ \times \int_K T_x(x|x(0), q, u(0))\sigma_0(x(0), q(0))\,dx(0)$$

without the normalizing constant. The main difficulty with solving for σ_1 is the fact that the integral is evaluated over K, as opposed to over $\mathbb{R}$. Because of the bounds on the integral, we cannot claim σ_n is Gaussian given that σ_{n-1} is. However, since the expression does quite closely resemble a Gaussian distribution, it may be possible to approximate σ_n by an unnormalized Gaussian distribution without losing significant accuracy. We intend to explore this possibility in future works. Further, we note that there may be classes of systems for which such straightforward sufficient statistics may be found.

4.6.2 Skid-Steered Vehicle

A skid-steered vehicle (SSV), modeled as a switched system, is presented in [CM11]. The SSV moves according to lateral sticking and sliding of its four wheels. [CM11] identify four modes associated with the vehicle: In mode 1, front and rear wheels stick laterally; in mode 2, front wheels stick and rear wheels skid laterally; in mode 3, front wheels skid and rear wheels stick; in mode 4, both front and rear wheels skid laterally. For each mode, the vehicle's continuous states X, Y, and θ are governed by a different set of second order ordinary differential equations (ODEs). The states X and Y represent the Cartesian coordinates for the vehicle's center of geometry, and θ gives the heading of the vehicle. We can represent the continuous state of the system by $x = (X, \dot{X}, Y, \dot{Y}, \theta, \dot{\theta})$, such that $\dot{x} = f_q(x) = f(x, q)$, with discrete state $q \in \mathcal{Q} = \{1, 2, 3, 4\}$. See [CM11] for expressions for f_q, which are too lengthy to reproduce here. We discretize the ODEs using an Euler approximation method to produce an equivalent discrete time system.

The control input can be expressed as a command informing the vehicle of what mode it should be in. If the vehicle responded perfectly, we would have $q_n = u_n$ for the mode at time n. Instead, let us assume that the mode changes behave similarly to the temperature regulation problem above, where the control command is implemented with a certain probability, dependent on the current mode: $T_q(q_{n+1}|q_n, u_n)$. The continuous state is assumed to be deterministic given the mode, so that $T_x(x_{n+1}|x_n, q_n, u_n) = 1_{f(x_n, q_n)}(x_{n+1})$. Note that we assume the continuous state updates first, then the discrete mode, so that the state x_{n+1} is dependent on mode q_n rather than q_{n+1}. Finally, assume we have a noisy observation of the continuous state, and have an observation of the mode which is not completely reliable.

$$
\begin{aligned}
y_n^x &= x_n + w_n \\
y_n^q &\sim Q_{q,y^q}
\end{aligned}
$$

The vector $w_n \in \mathbb{R}^6$ is an i.i.d. sequence of multivariate Gaussians with $w_n \sim \mathcal{N}(0, \mathcal{W})$. The matrix Q_{q,y^q} is given by

$$
Q_{q,y^q} = \begin{bmatrix} .9 & .033 & .033 & .033 \\ .033 & .9 & .033 & .033 \\ .033 & .033 & .9 & .033 \\ .033 & .033 & .033 & .9 \end{bmatrix}.
$$

Thus, the probability that the observed mode equals the true mode is 0.9, and if the observed mode is not the true mode, it is equally likely to be any of the other three modes.

The safe region K is defined as a path the vehicle should stay on. For instance, we could define K as a rectangular strip $K = \{X, Y : -1 \leq X \leq 1, \, -10 \leq Y \leq 10\}$. Assuming the initial position of the vehicle, x_0, is known and equal to $\hat{x}_0 \in K$, and the initial mode is independent of x_0, uniformly distributed, and represented by $\rho(q_0)$, σ_0 is given by

$$\sigma_0(x, q) = 1_{\hat{x}_0}(x)\rho(q) = \frac{1}{4} 1_{\hat{x}_0}(x).$$

In this case σ_1 is easily calculated.

$$\begin{aligned}\sigma_1(x, q) &= \sum_{q_0=1}^{4} \int_K 4Q_{q,y_1^q} T_q(q|q_0, u_0) \frac{\phi(y_1^x - x)}{\phi(y_1^x)} 1_{f(x_0,q_0)}(x)\sigma_0(x_0, q_0)\, dx_0 \\ &= \sum_{q_0=1}^{4} Q_{q,y_1^q} T_q(q|q_0, u_0) \frac{\phi(y_1^x - f(\hat{x}_0, q_0))}{\phi(y_1^x)} 1_{f(\hat{x}_0,q_0)}(x)\end{aligned}$$

Thus there are four possible values of x for which $\sigma_1(x, q)$ is nonzero. Similarly, given an x_n value, $\sigma_{n+1}(x, q)$ will only be nonzero for four values of x.

$$\sigma_{n+1}(x, q) = \sum_{q_n=1}^{4} \int_K 4Q_{q,y_{n+1}^q} T_q(q|q_n, u_n) \frac{\phi(y_{n+1}^x - f(x_n, q_n))}{\phi(y_{n+1}^x)} 1_{f(x_n,q_n)}(x)\sigma_n(x_n, q_n)\, dx_n \tag{4.42}$$

Even when σ_n takes the above seemingly simple form, there is no immediately obvious way to avoid evaluating the value functions for all $\sigma \in L^1$ in order to solve (4.26).

4.6.3 Computational Challenges

Because of the complexity of the hybrid dynamics and cost function, the sufficient statistic and dynamic programming equations are computationally intensive. The dynamic programming equations require iterating over an infinite state space. For the two examples presented here, one major challenge is circumventing the evaluation of the value functions for all $\sigma \in L^1(\mathcal{S})$ to solve (4.26). One possible alternative is using approximate dynamic programming to estimate the value functions V_n by sampling from y_n for each n to get sample trajectories of the σ_n. Since via our change of measure the y_n are i.i.d, such sampling should be straightforward. Each y_n^x is sampled from $\phi(\,\cdot\,)$, and each y_n^q is sampled from the uniform distribution on $\{1, \ldots, N_q\}$. Some work has explored approximate dynamic programming for DTSHS (see [KSS$^+$13]), in which the value function is approximated using a linear combination of basis functions, and constraints on the value function are evaluated by sampling from the state space. It is possible a similar approach could be applied to the partially observed case, where we must sample from the observation space to obtain instances of σ.

In addition, some work has been done on approximating continuous state POMDPs using point-based value iteration (see [PVSP06]), albeit in the context of additive cost functions with the belief state as a sufficient statistic. The method exploits the structure of the value function, and uses Monte Carlo methods to generate a set of samples from the belief space, in order to approximate the value function at a given starting belief state. Further, [BKLPR10] have applied this to a system with hybrid dynamics. These methods are the subject of the next chapter, and particularly how verification of a PODTSHS can be approximately solved by modifying existing algorithms for optimally controlling POMDPs.

4.7 Summary

We have presented a statistic sufficient for the control of a partially observable discrete time stochastic hybrid system, first for a viability objective, then for a viability objective with a change of measure, and finally for reach and reach-avoid objectives. By redefining the partially observed optimal control problem as one that is fully observed, with state variable σ (the information state generated by the sufficient statistic), we are able to define an optimal control policy as a function of σ. This control policy is equivalent to the policy defined as a function of the information vector, and leads to the same maximal safety probability. Further, we showed the equivalence between our approach and one that uses an additive cost function.

The major disadvantage of the sufficient statistic is that the dynamic programming equations must be solved for every possible $\sigma \in L^1$ at every time step. As a direct solution method, it is seemingly impractical. However, there may be cases where σ can be limited to a subset of L^1 so that the dynamic programming equations can be solved. Further, our choice of measure in defining the sufficient statistic may lend itself well to approximate dynamic programming techniques that avoid looping over all possible states. We present approximate solution strategies in the next chapter.

Chapter 5

Computational Methods for Reachability with Partial Observations

Although the sufficient statistic and dynamic program derived in the previous chapter provide a solution framework for the reachability problem under partial observations, they do not provide a practical means of generating probabilistic reachable sets. This chapter therefore focuses on numerical methods to approximately solve the reachability problem, based on the dynamic programming formulation of the previous chapter. We again focus on the viability problem in greatest detail, and explain how the methods for the viability problem extend to reachability and reach-avoid objectives. Two methods for approximately solving the dynamic program are presented. The first method approximates the stochastic hybrid system as an equivalent finite state Markov decision process, so that the information state is a probability mass function. The second approach approximates an indicator function over the safe region using Gaussian radial basis functions, to represent the information state as a Gaussian mixture. In both cases, we discretize the hybrid observation process and generate a sampled set of information states, then use point-based value iteration to under-approximate the viability probability. We obtain error bounds and convergence results in both cases, assuming additive Gaussian noise in the continuous state dynamics and observations. We compare the performance of the finite state and Gaussian mixture approaches on a simple numerical example.

5.1 Introduction

The focus of this chapter is on developing numerical approximation methods to the viability problem with partial observations. Building upon the work of the previous chapter, we exploit the relation between a PODTSHS and a POMDP, and the dynamic programming formulation we derived for the viability problem, to utilize established approximation methods for optimal control of POMDPs.

POMDPs are plagued by dimensionality on an even greater scale than MDPs. While MDPs suffer from the curse of dimensionality, and become intractable over state spaces of even moderate dimensions, the optimal control of a POMDP requires dealing with a so-called "curse of history" as well. Although a sufficient statistic may be used to condense the information provided in the history of observations and actions, the belief state produced by the sufficient statistic is a conditional probability function. Therefore, a dynamic programming solution that requires repeatedly solving a value function over all possible conditional probability distributions is clearly intractable. In particular, a continuous state space (e.g. $\mathcal{S} = \mathbb{R}^n$), requires a belief state that is a continuous function defined over an infinite domain (probability *density* function), and it is impossible to enumerate over all such functions. Therefore the study of efficient, approximate solutions to POMDPs is essential.

Although finding the solution to a general POMDP is hard [LGM01], many algorithms for approximating solutions to finite state POMDPs have been developed. These mainly rely on point-based value iteration (PBVI) schemes that only consider a subset of the belief space to update the value function used in the dynamic program. Because the value function is piecewise-linear and convex [Son71] (and so equivalently represented by a finite set of so-called α-vectors), sampling from the belief state provides a systematic way of storing a finite subset of those vectors. Such methods must be tailored to continuous state POMDPs because of the dimensionality of the belief state.

Therefore, as the first investigation into approximate probabilistic verification of PODTSHS, we formulate the viability problem for a PODTSHS as a partially observable Markov decision process with a non-additive cost function, and investigate representations of the belief state in either vector or Gaussian mixture form through finite- and continuous-state approximations to the PODTSHS. These representations allow us to exploit point-based methods developed for POMDPs, by sampling from the belief space and approximating the value function with a finite set of observations. All results for the viability problem can be readily extended to reachability and reach-avoid problems.

In this chapter we make several contributions to the solution of reachability problems for PODTSHS. First, we validate the use of POMDP solution techniques for viability analysis of a PODTSHS, by demonstrating that the value function is convex and admits an α-function representation similar to the piecewise-linear α-vector representation of a finite state POMDP. Second, we present a finite state approximation to the DTSHS that allows the belief state to take vector form under certain conditions, and show convergence for the approximation. Third, we preserve the continuity in the hybrid state space through a Gaussian mixture representation for the belief state, and approximate the indicator function that represents the safe region using Gaussian radial basis functions. In this case, we provide an error bound as a function of the L^1 error of the indicator function approximation. We outline a solution method that converges to the true solution from below, using either the finite or continuity-preserving belief state. Finally we demonstrate both approaches on a simple temperature regulation problem.

The rest of the chapter is organized as follows. Section 5.2 lists some of the related literature for solving POMDPs and viability problems for a perfectly observable DTSHS. Section 5.3 briefly reviews the viability problem for a PODTSHS, and Section 5.4 outlines PBVI for sub-optimal control of POMDPs. Section 5.5 justifies the use of POMDP solution techniques, and gives the finite and Gaussian mixture approximations to the viability problem for a PODTSHS as well as error bounds. Section 5.6 describes the use of point-based approximation techniques, through sampling of belief states and discretization of the observations. Section 5.8 provides preliminary numerical results, and Section 5.9 provides concluding remarks and directions for future work.

5.2 Related Work

To the best of our knowledge, this is the first work to provide computational results for verification of a PODTSHS. However, we draw upon related computational results for verification of a DTSHS, as well as approximation strategies from the POMDP community. We highlight some of the related work in both of these areas.

Computational results for analysis of perfectly observable stochastic hybrid systems are limited. Solutions via dynamic programming require evaluation of the value function over all possible states, which is infinite when those states are continuous. Discretization procedures can be employed to impose a finite number of states, as in [AAP+07], [AKLP10], and [SA13], which present rigorous uniform and adaptive gridding methods for verification of DTSHS.

A related approach is to generate an approximate abstraction of the original model (often in the form of a finite state system) that has the same properties [SA13], [FHH+11], [ZTA14]. The approximation is more easily solved than the original problem, and the approximate solution is directly related to the actual solution. Other solution strategies include approximate dynamic programming, where the value function of the dynamic program is approximated by a set of basis functions, as in [KSS+13]. Even so, current applications are limited to those with only a few discrete states and $\mathcal{X} \subseteq \mathbb{R}^n$ for n small.

Because an exact solution is often impossible to obtain for even a finite state POMDP, approximation strategies have been studied extensively. Many variants of

PBVI are available (e.g. [PGT06], [KHL08], [NJ00], [SV05]) that generally differ in how they sample from the belief space (for example [SPK13] for an overview). More recently, approximate solutions for continuous-state POMDPs have been studied (for example [Thr00], [BMWDW06], [ZFM10], [PTKLP10], [ES10], [PTKLP11]).

Often the continuous state can be discretized and approximately solved as a finite state POMDP using PBVI methods. Depending on the dimensionality and behavior of the system, it may be more informative or computationally more efficient to preserve the continuity of the state space. Many existing methods for continuous state POMDPs assume the belief state is Gaussian (e.g. [BMWDW06], [ZFM10]), and represent the belief state in a parameterized form which is then discretized and solved as a finite state MDP. When the belief state cannot adequately be represented using a single Gaussian, a Gaussian mixture may be used instead. An equivalent point-based algorithm for continuous-state POMDPs using Gaussian mixtures is presented in [PVSP06], and demonstrated on a stochastic hybrid system with hidden modes in [BKLPR10].

The results of this chapter are mainly inspired by the discretization procedure of [AKLP10] and the Gaussian mixture formulation of [PVSP06].

5.3 Problem Formulation

The problem we wish to solve is similar to that of Chapter 4. We again consider a PODTSHS (Definition 4.1) with hybrid state space $\mathcal{S} = \mathcal{X} \times \mathcal{Q}$, observation space $\mathcal{Y} = \mathcal{Y}^x \times \mathcal{Y}^q$, finite control space $\mathcal{U}$, stochastic transition function $\tau_s(ds'|s, u)$, stochastic observation function $\tau_y(dy|s, u)$, and initial density ρ on $\mathcal{S}$. We make some additional simplifying assumptions on the state transitions and observation process, although as pointed out later these are not necessary for all subsequent results to hold.

In what follows we assume the continuous state x obeys the affine difference equation

$$x_{n+1} = Ax_n + g(q_n, u_n, q_{n+1}) + v_n. \tag{5.1}$$

Matrix A is assumed invertible, and v_n are independent and identically distributed Gaussian random variables for all n, $v_n \sim \mathcal{N}(0, \mathcal{V})$. Therefore kernel T_x admits a Gaussian density, with $T_x(dx' \in B|q', x, q, u) = \int_B \phi(dx'; Ax + g(q, u, q'), \mathcal{V})$. The function ϕ is used to represent a Gaussian probability density function (pdf); $\phi(x; \mu, \Sigma)$ is equal to a Gaussian pdf with mean μ and covariance Σ evaluated at x.

We assume that the continuous observation y_n^x is equal to the state x_n, disrupted by additive Gaussian noise $w_n \sim \mathcal{N}(0, \mathcal{W})$.

$$y_n^x = x_n + w_n \tag{5.2}$$

The observation function T_{y^x} therefore also has a Gaussian density, with $T_{y^x}(dy \in B|x) = \int_B \phi(dy; x, \mathcal{W})$. The discrete observation space is defined for simplicity as $\mathcal{Y}^q = \mathcal{Q}$.

Finally, the initial density ρ is assumed Gaussian in x: $\rho(x, q) = Q_0(q)\phi(x; \mu_0, \Sigma_0)$ such that $\sum_{q \in \mathcal{Q}} \int_{\mathcal{X}} \rho(x, q)\, dx = 1$.

Because T_{y^x}, T_x, and ρ are Gaussian, and $\mathcal{U}$ is finite and bounded, the following Lipschitz properties hold.

$$\begin{aligned}
\|T_x(x'|q', s, u) - T_x(\overline{x}'|q', s, u)\| &\leq h_x^{(1)} \|x' - \overline{x}'\| \\
\|T_x(x'|q', x, q, u) - T_x(x'|q', \overline{x}, q, u)\| &\leq h_x^{(2)} \|x - \overline{x}\| \\
\|T_{y^x}(y|x) - T_{y^x}(\overline{y}|x)\| &\leq h_y^{(1)} \|y - \overline{y}\| \\
\|T_{y^x}(y|x) - T_{y^x}(y|\overline{x})\| &\leq h_y^{(2)} \|x - \overline{x}\|
\end{aligned} \tag{5.3}$$

We define the maximum values of the densities associated with T_x and T_{y^x} as ϕ_v^* and ϕ_w^*, respectively, with $\phi_v^* = (2\pi)^{-\frac{n}{2}} |\mathcal{V}|^{-\frac{1}{2}}$ and $\phi_w^* = (2\pi)^{-\frac{n}{2}} |\mathcal{W}|^{-\frac{1}{2}}$.

We focus on Problem 2.1 of Chapter 2 in the case of partial observations, and so would like to find

$$p^N_{\text{viab}}(\rho; K) = \sup_{\pi \in \Pi} \mathbb{E}^{\pi} \left[\prod_{n=0}^{N} 1_K(s) \middle| \rho \right], \tag{5.4}$$

$$\pi^* = \arg\sup_{\pi \in \Pi} \left\{ p^N_{\text{viab}}(\rho; K) \right\} \tag{5.5}$$

which is formally stated as

Problem 5.1. *Consider a PODTSHS $\mathcal{H}^{po}$ (defined in Definition 4.1). Given a safe set K and time horizon N we would like to:*

1. *Compute the maximal probability (5.4) of remaining within K for N time steps.*
2. *Compute the optimal policy π^* given by (5.5).*

If the maximal probability and optimal policy cannot be computed exactly (which is quite likely [LGM01]), an approximation that produces a suboptimal policy and lower bound on the maximal viability probability is desired.

5.4 Approximation Strategies for POMDPs

For a finite state POMDP $\mathcal{J}^{po}$ (see Definition 4.2), Sondik [Son71] first showed that over a finite horizon $N < \infty$, the value function (4.8) at each time n is piecewise-linear and convex in the belief state, and thus can be expressed as

$$V^*_n(b) = \max_{\alpha^i_n \in \Gamma_n} \sum_s \alpha^i_n(s) b(s). \tag{5.6}$$

The functions $\alpha^i_n \in \Gamma_n$, or "α-vectors", represent a policy tree that starts from a specific action u and state s, and specifies optimal actions conditioned on observations for time steps $n+1$ to N. The α-vectors thus characterize the current value of being in state s and taking action u, plus the expected sum of future rewards assuming all subsequent actions are chosen optimally. Because each α-vector is associated with a specific action, by picking the α-vector that maximizes $\sum_s \alpha^i_n(s) b(s)$, we also define the optimal policy for belief b at time n.

Calculating the value function and optimal policy requires knowledge of the complete sets of α-vectors, Γ_n, for all n. The α-vectors at time n are computed recursively from the α-vectors calculated at time $n+1$. For each action, we observe one of $|\mathcal{Y}|$ observations (where $|\cdot|$ indicates the cardinality of the set), and for each observation there is a subsequent α-vector defined at time $n+1$, resulting in $|\mathcal{U}||\Gamma_{n+1}|^{|\mathcal{Y}|}$ α-vectors at time n. Hence using the α-vector representation to optimize a POMDP is often infeasible, because the number of α-vectors grows exponentially.

An approximate solution can be obtained though point-based value iteration (PBVI), in which a lower bound to the value function is computed using a finite subset $B \subset \mathcal{B}$, the set of all belief states. The general idea is to generate a collection of points $b \in B$, and for each of these points, estimate the value function via a "backup" operation. PBVI approaches are distinguished by how B is selected [PGT06], [SPK13].

We summarize the most common method of under-estimating the value function, assuming B has already been selected. One α-vector is generated for each belief point $b^i \in B$, $B = (b^0, b^1, \dots, b^m)$, so that $\tilde{\Gamma}_n = (\alpha_n^0, \alpha_n^1, \dots, \alpha_n^m)$ for all n. We assume that an α-vector α_n^j corresponding to b^j will apply to all belief points in a region around b^j (i.e. for any b in a neighborhood of b^j the same action will likely be optimal). Hence the value at some b not necessarily in B can be approximated by $V_n^*(b) \approx \max_{\alpha_n^i \in \tilde{\Gamma}_n} \sum_s \alpha_n^i(s) b(s)$ as in (5.6) but with a restricted set $\tilde{\Gamma}_n \subset \Gamma_n$. The set $\tilde{\Gamma}_n$ is generated recursively from $\tilde{\Gamma}_{n+1}$, but without enumeration over all possible combinations of observations and subsequent α-vectors in $\tilde{\Gamma}_{n+1}$, by using the following backup operation for each $b \in B$.

$$backup(b) = \arg\max_{\alpha_n^i \in \tilde{\Gamma}_n} \sum_{s \in \mathcal{S}} \alpha_n^i(s) b(s) \tag{5.7}$$

The overall PBVI algorithm then consists of selecting a set of belief points B, and repeatedly applying (5.7) to each element of B. In the case of a finite horizon of length N, the backup operator will be applied N times, and for an infinite horizon, the backup operator will be applied until some tolerance level is reached (for example, where $\|V_{n+1}(b) - V_n(b)\| < \epsilon$).

The above derivations apply to a model with discrete state, action, and observation spaces, but can be extended to POMDPs with a continuous state space and discrete observation and action spaces [PVSP06]. In this case, the α-vectors are replaced by α-functions defined over the continuous space S. Because the observations and actions are discrete, the number of α-functions is finite, and the value function is piecewise-linear and convex under the α-function representation. Further, summations over $\mathcal{S}$ are replaced by integrals, hence (5.7) is written as $backup(b) = \arg\max_{\alpha_n^i \in \tilde{\Gamma}_n} \langle \alpha_n^i, b \rangle$. We maintain this notation in our derivations, where in the case of a hybrid state space with continuous state x and discrete state q, $\langle f, g \rangle = \sum_q \int f(x,q) g(x,q)\, dx$ for well-defined functions f and g.

5.5 Reformulation of Problem 5.1

We exploit PBVI to solve Problem 5.1, by transforming it into an optimal control problem for a POMDP. Hence we first restate the dynamic programming formulation for the viability problem given $\mathcal{H}^{po}$ from Chapter 4. We then show that the value function admits an α-function representation, that the α-functions and belief states can be approximately represented in closed form (as either vectors or Gaussian mixtures), and that finite collections of each may be generated and used to approximate (5.4), similar to a point-based POMDP solver.

We present two approximations of Problem 5.1 for the PODTSHS $\mathcal{H}^{po}$. The first discretizes $\mathcal{S}$ to produce a finite state POMDP, and the second preserves continuity in $\mathcal{S}$ by using a Gaussian mixture approach, thus characterizing the PODTSHS by a collection of weights, means, and covariances.

5.5.1 Validity of POMDP Formulation

We restate the sufficient statistic $\eta = (\eta_0, \dots, \eta_N)$ and value function for Problem 5.1 presented in Chapter 4.

$$\eta_n(\rho, i_n) = \mathbb{E}^\pi \left[1_q(q_n) 1_x(x_n) \prod_{i=0}^{n-1} 1_K(s_i) \middle| \rho, i_n \right] \tag{5.8}$$

Recall the information state $\sigma_n(x_n, q_n) \in \Sigma \subseteq L^1(\mathcal{S})$ (where $L^1(\mathcal{S})$ is the space of integrable functions defined on $\mathcal{S}$) such that $\eta_n(\rho, i_n) = \sigma_n$, which is distinct from the belief state (e.g. the conditional distribution of the current state).

$$\begin{aligned} \sigma_0 &= \rho \\ \sigma_n &= \Phi_{y_n, u_{n-1}} \sigma_{n-1} \end{aligned} \tag{5.9}$$

where $\Phi_{y,u}\sigma$ is given by

$$\left(\Phi_{y,u}\sigma\right)(s') = \frac{1}{\mathbb{P}(y|\sigma, u)} \tau_y(y|s', u) \int_K \tau_s(s'|s, u) \sigma(s)\, ds. \tag{5.10}$$

The dynamic programming recursion over σ is

$$\begin{aligned} V_N^*(\sigma) &= \langle \sigma, 1_K \rangle \\ V_n^*(\sigma) &= \max_{u \in \mathcal{U}} \mathbb{E}^\pi \left[V_{n+1}^*(\Phi_{y,u}\sigma) \right] \end{aligned} \tag{5.11}$$

with solution $V_0^*(\rho) = p_{\text{viab}}^N(\rho; K)$. The optimal policy is $\pi^* = (\mu_0^*, \dots, \mu_{N-1}^*)$, with

$$\mu_n^*(\sigma_n) = \arg\max_{u \in \mathcal{U}} V_n^*(\sigma_n) \tag{5.12}$$

for all $n \in [0, N]$.

Lemma 5.1. *For any n, the value function (5.11) can be written as*

$$V_n^*(\sigma) = \sup_{\alpha_n^i \in \Gamma_n} \langle \alpha_n^i, \sigma \rangle.$$

Proof. By induction. At time N,

$$V_N^*(\sigma) = \int_{\mathcal{S}} 1_K(s) \sigma(ds).$$

By defining $\alpha_N(s) = 1_K(s)$, we obtain the desired result. Note that this definition of α_N is in line with the definition given in Section 5.4, because although it does not represent a full policy tree (being at the terminal time, there are no more branches on the tree), it does represent the immediate value of being in state (x, q), given by $1_K(x, q)$.

Next, assuming $V^*_{n+1}(\sigma) = \sup_{\Gamma_{n+1}} \langle \alpha^i_{n+1}, \sigma \rangle$, V^*_n can be written as

$$\begin{aligned} V^*_n(\sigma) &= \max_{u \in \mathcal{U}} \mathbb{E}^\pi \left[\sup_{\alpha^i_{n+1} \in \Gamma_{n+1}} \langle \alpha^i_{n+1}, \sigma \rangle \right] \\ &= \max_{u \in \mathcal{U}} \int_{\mathcal{Y}} \sup_{\alpha^i_{n+1} \in \Gamma_{n+1}} \langle \alpha^i_{n+1}, \Phi_{y,u}\sigma \rangle \mathbb{P}(dy|\sigma, u) \\ &= \max_{u \in \mathcal{U}} \int_{\mathcal{Y}} \sup_{\alpha^i_{n+1} \in \Gamma_{n+1}} \left[\int_K \int_{\mathcal{S}} \alpha^i_{n+1}(s') \tau_y(y|s', u) \tau_s(ds'|s, u) \sigma(ds) \right] dy. \end{aligned}$$

Then for a specific observation y, action u, and α^i_{n+1} function, the function $\alpha^i_{y,u}$ can be defined as

$$\alpha^i_{y,u}(s) = \int_{\mathcal{S}} \alpha^i_{n+1}(s') \tau_y(y|s', u) \tau_s(ds'|s, u) 1_K(s). \tag{5.13}$$

Because $\alpha^i_{y,u}$ does not depend on σ, we can redefine the supremum over all Γ_{n+1} to be over all $\alpha^i_{y,u}$.

$$V^*_n(\sigma) = \max_{u \in \mathcal{U}} \int_{\mathcal{Y}} \sup_{\{\alpha^i_{y,u}\}} \langle \alpha^i_{y,u}, \sigma \rangle \, dy^x$$

For a specific σ, u, and y, we define

$$\begin{aligned} \alpha_{y,u,\sigma}(s) &= \arg\sup_i \langle \alpha^i_{y,u}, \sigma \rangle \\ &= \int_{\mathcal{S}} \alpha^{*(y)}_{n+1}(s') \tau_y(y|s', u) \tau_s(ds'|s, u) 1_K(s) \end{aligned} \tag{5.14}$$

with $*(y)$ denoting the index i of the α-function $\alpha^i_{y,u}$ that maximizes the inner product. We further simplify V^*_n as

$$\begin{aligned} V^*_n(\sigma) &= \max_{u \in \mathcal{U}} \int_{\mathcal{Y}} \langle \alpha_{y,u,\sigma}, \sigma \rangle \, dy \\ &= \max_{u \in \mathcal{U}} \left\langle \int_{\mathcal{Y}} \alpha_{y,u,\sigma} \, dy, \sigma \right\rangle. \end{aligned} \tag{5.15}$$

Therefore, the set of all $\{\alpha_n^i\}$ is described by

$$\Gamma_n = \bigcup_{\sigma} \left\{ \int_{\mathcal{Y}} \alpha_{y,u^*,\sigma} \, dy \right\} \tag{5.16}$$

with u^* the control inputs chosen according to (5.15), and V_n^* may be written as

$$V_n^*(\sigma) = \sup_{\alpha_n^i \in \Gamma_n} \langle \alpha_n^i, \sigma \rangle \tag{5.17}$$

■

Lemma 5.2. *The value function* (5.11) *is convex in* σ *for all* $n = 0, \ldots, N$, $\sigma, \overline{\sigma} \in L^1$ *and* $0 \leq \lambda \leq 1$*:*

$$V_n^*(\lambda\sigma_1 + (1-\lambda)\sigma_2) \leq \lambda V_n^*(\sigma_1) + (1-\lambda)V_n^*(\sigma_2).$$

Proof. The proof follows directly from Lemma 5.1. Because the inner product operator is linear, and Lemma 5.1 states that

$$V_n^*(\sigma) = \sup_{\alpha_n^i \in \Gamma_n} \langle \alpha_n^i, \sigma \rangle,$$

it follows directly that

$$\begin{aligned} V_n^*(\lambda\sigma + (1-\lambda)\overline{\sigma}) &= \sup_{\alpha_n^i \in \Gamma_n} \langle \alpha_n^i, \lambda\sigma + (1-\lambda)\overline{\sigma} \rangle \\ &= \langle \alpha_n^*, \lambda\sigma + (1-\lambda)\overline{\sigma} \rangle \\ &= \lambda \langle \alpha_n^*, \sigma \rangle + (1-\lambda)\langle \alpha_n^*, \overline{\sigma} \rangle \\ &\leq \sup_{\alpha_n^i \in \Gamma_n} \lambda \langle \alpha_n^i, \sigma \rangle + \sup_{\alpha_n^i \in \Gamma_n} (1-\lambda)\langle \alpha_n^i, \overline{\sigma} \rangle \\ &\leq \lambda V_n^*(\sigma) + (1-\lambda)V_n^*(\overline{\sigma}). \end{aligned}$$

■

Since Lemmas 5.2 and 5.1 show that the value function (5.11) is convex and admits an α-function representation, $\mathcal{H}^{po}$ is amenable to POMDP solution techniques. Note that Lemma 5.1 is not useful for solving Problem 5.1 directly, since Γ_n is not finite and the α-functions and information states have no common structure. If we assume a finite number of observations and control inputs, i.e. $|\mathcal{Y}| < \infty$ and $|\mathcal{U}| < \infty$, then the value function is piecewise-linear and convex, and the size of Γ_n (number of α-functions stored at time n, $|\Gamma_n|$) is finite. However, we do not make this assumption, and will require additional approximation strategies, discussed in Section 5.6.

5.5.2 Finite State Approximation

We first consider a finite state POMDP [AKLP10], whose solution converges to the true viability probability (5.4) and optimal policy (5.5). The state space $\mathcal{S}$ is discretized to obtain a vector representation of α and σ. The observation space is unchanged (i.e. hybrid), because the set of observations only affects the finiteness of sets Γ_n and Σ_n. We defer discussion of producing finite collections of Γ_n and Σ_n to section 5.6.

Given compact safe set $K \in \mathcal{B}(\mathcal{S})$, let $K = \bigcup_{q \in \mathcal{Q}} K_q \times \{q\}$. Denote $\lambda = \max_{q \in \mathcal{Q}} \mathcal{L}(K_q)$, the finite Lebesgue measure of $K_q \subset \mathbb{R}^n$. Each K_q is partitioned into a finite number of subsets, so that $K_q = \bigcup_{i=1}^{m_q} K_{i,q}$, with $K_{i,q}$ pairwise disjoint (i.e. $K_{i,q} \cap K_{j,q} = \emptyset$ for all $i \neq j$), $K_{i,q} \in \mathcal{B}(\mathbb{R}^n)$. Finally, let $\delta^x_{i,q}$ be the diameter of partition $K_{i,q}$ so that $\delta^x_{i,q} = \sup\{\|x - \overline{x}\| : x, \overline{x} \in K_{i,q}\}$, with $\delta^x = \max_{i,q} \delta^x_{i,q}$.

The partition of K is denoted by $\mathcal{G}^s = \bigcup_{i=1,\ldots,m_q, q \in \mathcal{Q}} \mathcal{G}^s_{i,q}$ with $\mathcal{G}^s_{i,q} = K_{i,q} \times \{q\}$. Each element $\mathcal{G}^s_{i,q}$ has a representative point $(x^{i,q}, q)$ and the set $K_\delta = \{(x^{i,q}, q) : i = 1, \ldots, m_q, q \in \mathcal{Q}\}$ is the discrete representation of K. We do not consider here how the points $(x^{i,q}, q)$ are selected, but an example is provided in Section 5.8. The function $\xi : K \to K_\delta$ maps a state $s \in \mathcal{G}^s_{i,q}$ to its representative point $(x^{i,q}, q)$ and the function $\Xi : K_\delta \to K$ is the set-valued map from discrete point $(x^{i,q}, q)$ to its corresponding set $\mathcal{G}^s_{i,q}$. The discrete state space is defined as $Z_\delta = K_\delta \bigcup \{\psi_s\}$, with ψ_s a single variable that represents all states $s \in \mathcal{S} \backslash K$.

Definition 5.1. *(POMDP approximation to PODTSHS, $\hat{\mathcal{H}}^{po}$). The POMDP approximation is a tuple $\hat{\mathcal{H}}^{po} = (Z_\delta, \mathcal{Y}, \mathcal{U}, \tau^\delta_s, \tau_y, \rho_\delta)$ where*

1. *Z_δ is a finite set of discrete states*

2. *$\mathcal{Y}$ is as defined in Definition 1*

3. *$\mathcal{U}$ is as defined in Definition 1*

4. *$\tau_s^\delta : Z_\delta \times \mathcal{U} \times Z_\delta \to [0,1]$ is a discrete state transition function that assigns probabilities to elements of Z_δ*

5. *τ_y is as defined in Definition 1*

6. *$\rho_\delta : Z_\delta \to [0,1]$ is a function that assigns probabilities to elements of Z_δ at time zero*

We define the transition function as

$$\tau_s^\delta(z'|z,u) = \begin{cases} \tau_s(\Xi(z')|z,u), & \text{if } z' \in K_\delta \text{ and } z \in K_\delta \\ 1 - \sum_{\overline{z}\in K_\delta} \tau_s(\Xi(\overline{z})|z,u), & \text{if } z' = \psi_s \text{ and } z \in K_\delta \\ 1, & \text{if } z' = \psi_s \text{ and } z = \psi_s \\ 0, & \text{if } z' \in K_\delta \text{ and } z = \psi_s \end{cases} \tag{5.18}$$

with $\sum_{z'\in Z_\delta} \tau_s^\delta(z'|z,u) = 1$, and the initial distribution ρ_δ on Z_δ as

$$\rho_\delta(z) = \begin{cases} \rho(\Xi(z)), & \text{if } z \in K_\delta \\ 1 - \sum_{\overline{z}\in K_\delta} \rho(\Xi(\overline{z})) & \text{if } z = \psi_s \end{cases} \tag{5.19}$$

Recall that $\tau_s(\Xi(z')|z,u) = T_x(K_{i,q'}|q',z,u)T_q(q'|q,u)$, and T_x evaluated over Borel set $K_{i,q'}$ is a Gaussian density integrated over set $K_{i,q'}$. The discrete probability space is $(\Omega_\delta, \sigma(\Omega_\delta), \mathbb{P}_\delta^{\pi_\delta})$ with $\Omega_\delta = Z_\delta^{N+1} \times \mathcal{Y}^N$, $\sigma(\Omega_\delta)$ the σ-algebra on Ω_δ, and $\mathbb{P}_\delta^{\pi_\delta}$ the probability measure uniquely defined by ρ_δ, τ_y, τ_s^δ, and a control policy $\pi_\delta = (\mu_0^\delta, \ldots, \mu_{N-1}^\delta)$, $\mu_n^\delta : \Sigma_\delta \to \mathcal{U}$, with Σ_δ the set of all information states σ_δ.

We further define the operator $\Phi_{y,u}^\delta$ and the intermediate vector $\alpha_{y,u,\sigma_\delta}^\delta$ as

$$\left(\Phi_{y,u}^\delta \sigma_\delta\right)(z') = \frac{1}{\mathbb{P}(y|\sigma_\delta,u)} \tau_y(y|z',u) \sum_{z\in K_\delta} \tau_s^\delta(z'|z,u)\sigma_\delta(z) \tag{5.20}$$

$$\alpha_{y,u,\sigma_\delta}^\delta(z) = \sum_{z'\in K_\delta} \alpha_{n+1,\delta}^{*(y)}(z')\tau_y(y|z',u)\tau_s^\delta(z'|z,u)1_{K_\delta}(z) \tag{5.21}$$

for $y \in \mathcal{Y}$, $u \in \mathcal{U}$, $z', z \in Z_\delta$. The viability problem for $\hat{\mathcal{H}}^{po}$ is

$$\sup_{\pi_\delta \in \Pi_\delta} p^N_{\text{viab}}(\pi_\delta, \rho_\delta; K_\delta) = \sup_{\pi_\delta \in \Pi_\delta} \mathbb{P}^{\pi_\delta}_\delta \left[z_n \in K_\delta,\ \forall\, n \in [0, N] \right]. \tag{5.22}$$

To solve (5.22), we formulate the information state σ_δ and the value function $V^*_{n,\delta} : \sigma_\delta \to [0,1]$ for $n = 0, \ldots, N$.

The discrete information state represents a probability mass function over Z_δ, and can be expressed as an integral over an equivalent density (just as $\tau^\delta_s(z'|z,u) = \tau_s(\Xi(z')|z,u)$)

$$\sigma_{n,\delta}(z) = \begin{cases} \int_{\Xi(z)} \hat{\sigma}_n(ds), & \text{if } z \in K_\delta \\ \int_{\mathcal{S} \setminus K} \hat{\sigma}_n(ds), & \text{if } z = \psi_s \end{cases} \tag{5.23}$$

with $\hat{\sigma}_n(s)$ given by

$$\hat{\sigma}_n(s') = \begin{cases} \rho(s'), & \text{if } n = 0 \\ \left(\hat{\Phi}_{y,u} \hat{\sigma}_{n-1} \right)(s') = \dfrac{1}{\mathbb{P}(y|\hat{\sigma}_{n-1}, u)} \tau_y(y|\xi(s'), u) & \\ \qquad \times \displaystyle\int_K \tau_s(s'|\xi(s), u) \hat{\sigma}_{n-1}(ds), & \text{if } n > 0 \end{cases} \tag{5.24}$$

This can be verified by substituting the expression for τ^δ_s in terms of τ_s into (5.20) and using an induction argument.

The value function is

$$\begin{aligned} V^*_{N,\delta}(\sigma_\delta) &= \sum_{z \in K_\delta} \sigma_\delta(z) \\ V^*_{n,\delta}(\sigma_\delta) &= \max_{u \in \mathcal{U}} \int_{\mathcal{Y}} V^*_{n+1,\delta}(\Phi^\delta_{y,u} \sigma_\delta) \mathbb{P}(dy|\sigma_\delta, u) \end{aligned} . \tag{5.25}$$

The maximum probability of remaining within K_δ over N time steps (5.22) is

$$p^N_{\text{viab}}(\rho_\delta; K_\delta) = V^*_{0,\delta}(\rho_\delta). \tag{5.26}$$

We now show that the viability probability for the finite state approximation $\hat{\mathcal{H}}^{po}$ converges to the true solution as grid size parameter δ^x tends to zero. To do so, we first describe the error between the continuous information state σ and the vector approximation σ_δ.

Information State Approximation Error

We first characterize the relationship between the densities σ (5.9) and $\hat{\sigma}$ (5.24) in the following theorem.

Theorem 5.1. *The density $\hat{\sigma}$ defined in* (5.24) *satisfies*

$$|\sigma_n(s) - \hat{\sigma}_n(s)| \leq \eta_n^\sigma \delta^x$$

for all $s \in \mathcal{S}$, $\sigma_n \in \Sigma$, and η_n^σ given by

$$\eta_n^\sigma = \sum_{i=1}^{n} c_{1,i} \left(\prod_{j=i+1}^{n} c_{2,j} \right),$$

with $c_{1,i} = \min\left\{ \frac{1}{\mathbb{P}(y|\sigma_n, u)}, \frac{1}{\mathbb{P}(y|\hat{\sigma}_n, u)} \right\} [\phi_v^* h_y^{(2)} + \phi_w^* h_x^{(2)}]$, *and*
$c_{2,j} = \min\left\{ \frac{1}{\mathbb{P}(y|\sigma_n, u)}, \frac{1}{\mathbb{P}(y|\hat{\sigma}_n, u)} \right\} \phi_w^* N_q \lambda.$

Proof. By induction. At time $n = 0$, $\sigma_0(s) = \rho(s) = \hat{\sigma}_0(s)$ and the inequality is trivially satisfied. For all $i = 0, \ldots, n$, assume $|\sigma_i(s) - \hat{\sigma}_i(s)| \leq \eta_i^\sigma \delta^x$. At time $i = n+1$, for any $y \in \mathcal{Y}$ and any $u \in \mathcal{U}$,

$$\begin{aligned} |\sigma_{n+1}(s') - \hat{\sigma}_{n+1}(s')| \leq \min&\left\{ \frac{1}{\mathbb{P}(y|\sigma_n, u)}, \frac{1}{\mathbb{P}(y|\hat{\sigma}_n, u)} \right\} \\ &\times \Bigg| \tau_y(y|s', u) \int_K \tau_s(s'|s, u)\sigma_n(ds) \\ &\qquad - \tau_y(y|\xi(s'), u) \int_K \tau_s(s'|\xi(s), u)\hat{\sigma}_n(ds) \Bigg| \end{aligned} \tag{5.27}$$

We add and subtract $\tau_y(y|\xi(s'), u) \int_K \tau_s(s'|s, u)\sigma_n(ds)$ and
$\tau_y(y|\xi(s'), u) \int_K \tau_s(s'|\xi(s), u)\sigma_n(ds)$ and apply the triangle inequality.

$$\begin{aligned} |\sigma_{n+1}(s') - \hat{\sigma}_{n+1}(s')| \leq \min&\left\{ \frac{1}{\mathbb{P}(y|\sigma_n, u)}, \frac{1}{\mathbb{P}(y|\hat{\sigma}_n, u)} \right\} \\ &\times \Bigg[\left|\tau_y(y|s', u) - \tau_y(y|\xi(s'), u)\right| \int_K \tau_s(s'|s, u)\sigma_n(ds) \\ &\quad + \tau_y(y|\xi(s'), u) \int_K \left|\tau_s(s'|s, u) - \tau_s(s'|\xi(s), u)\right| \sigma_n(ds) \\ &\quad + \tau_y(y|\xi(s'), u) \int_K \tau_s(s'|\xi(s), u) \left|\sigma_n(s) - \hat{\sigma}_n(s)\right| ds \Bigg] \end{aligned}$$

$$\begin{aligned}
|\sigma_{n+1}(s') - \hat{\sigma}_{n+1}(s')| \leq \min\left\{\frac{1}{\mathbb{P}(y|\sigma_n, u)}, \frac{1}{\mathbb{P}(y|\hat{\sigma}_n, u)}\right\} &\sup_{s'\in K} \Big[T_{y^q}(y^q|q', u) \\
&\times \left|T_{y^x}(y^x|x') - T_{y^x}(y^x|\xi(x'))\right| \phi_v^* + \phi_w^* T_q(q'|q,u) \\
&\times \left|T_x(x'|q', s, u) - T_x(x'|q', \xi(s), u)\right| \\
&+ \phi_w^* \|\sigma_n - \hat{\sigma}_n\|_\infty \int_K \tau_s(s'|\xi(s), u)\, ds \Big] \\
\leq \min\left\{\frac{1}{\mathbb{P}(y|\sigma_n, u)}, \frac{1}{\mathbb{P}(y|\hat{\sigma}_n, u)}\right\} & \\
+ \Bigg[h_y^{(2)} \|x' - \xi(x')\| \phi_v^* &+ \phi_w^* h_x^{(2)} \|x' - \xi(x')\| \\
+ \phi_w^* \|\sigma_n - \hat{\sigma}_n\|_\infty &\sum_{q\in\mathcal{Q}} T_q(q'|q,u) \int_{K_q} T_x(x'|\xi(s), u)\, dx \Bigg]
\end{aligned} \tag{5.28}$$

Since T_x is bounded by ϕ_v^*, and the Lebesgue measure of K_q is at most λ, we obtain

$$\begin{aligned}
|\sigma_{n+1}(s') - \hat{\sigma}_{n+1}(s')| &\leq \min\left\{\frac{1}{\mathbb{P}(y|\sigma_n, u)}, \frac{1}{\mathbb{P}(y|\hat{\sigma}_n, u)}\right\} \\
&\qquad \times \left[h_y^{(2)} \phi_v^* \delta^x + \phi_w^* h_x^{(2)} \delta^x + \phi_w^* N_q \lambda \|\sigma_n - \hat{\sigma}_n\|_\infty \right] \\
&\leq \min\left\{\frac{1}{\mathbb{P}(y|\sigma_n, u)}, \frac{1}{\mathbb{P}(y|\hat{\sigma}_n, u)}\right\} \\
&\qquad \times \left[h_y^{(2)} \phi_v^* \delta^x + \phi_w^* h_x^{(2)} \delta^x + \phi_w^* N_q \lambda \left(\eta_n^\sigma \delta^x\right) \right].
\end{aligned}$$

Combining terms gives the desired result. ■

Convergence of (5.26) to (5.4)

The value function (5.11) requires integrating over spaces $\mathcal{Y}$ and $\mathcal{S}$ of unbounded size. To prove convergence of the value function $V_{n,\delta}^*$ to V_n^*, we must show that these integrals are bounded.

Consider the following two lemmas regarding integration of T_{y^x} and T_x over unbounded sets $\mathcal{Y}^x$ and $\mathcal{X}$, respectively.

Lemma 5.3. *For any $x, \overline{x} \in K_{i,q}$, for all $i = 1, \ldots, m_q$, $q \in \mathcal{Q}$, the following holds:*

$$\int_{\mathcal{Y}^x} |T_{y^x}(y^x|x) - T_{y^x}(y^x|\overline{x})|\, dy^x \leq \left[\beta_{1,i}^y h_y^{(2)} + \beta_2^y\right] \delta_{i,q}^x$$

with $\beta^y_{1,i} = \int_{\{y^x : \|y^x - x\|_2^2 \leq w^*_\sigma, y^x \in \mathcal{Y}^x, x \in K_{q,i}\}} 1 \, dy^x$ *and* $\beta^y_2 = \frac{\phi^*_w}{2}\sqrt{2w^*_\sigma \pi}$*. In other words,* $\beta^y_{1,i}$ *is the Lebesgue measure of region* $K_{i,q}$ *scaled by* $\sqrt{w^*_\sigma}$ *in all directions, with* w^*_σ *the largest singular value of* $\mathcal{W}$ *(so if* $K_{i,q}$ *is a hyperrectangle, each side will increase by a factor of* $2\sqrt{w^*_\sigma}$*).*

Proof. We exploit properties of the derivative of a Gaussian distribution, which bounds the Lipschitz constants for T_{y^x} from above. The constant $h^{(2)}_y$ is the maximum value of the derivative of $\phi(y^x; x, \mathcal{W})$ with respect to x:

$$\left\| \frac{\partial \phi}{\partial x} \right\|_2 \leq \frac{1}{(2\pi)^{\frac{n}{2}} |\mathcal{W}|^{\frac{1}{2}} w^*_\sigma} \|x - y^x\|_2 e^{-\frac{\|x-y^x\|_2^2}{2w^*_\sigma}}. \tag{5.29}$$

Since $\|\mathcal{W}\|_2 = |\rho(\mathcal{W})| = w^*_\sigma$ (with $\rho(\mathcal{W})$ the largest eigenvalue of $\mathcal{W}$), the maximum of (5.29) occurs at $\|x - y^x\| = \sqrt{w^*_\sigma}$.

Although $\|\frac{\partial \phi}{\partial x}\| \leq h^{(2)}_y$, we create a tighter bound for the case in which $\|x - y^x\|$ is greater than $\sqrt{w^*_\sigma}$ (for $y^x \in \mathcal{Y}^x$ such that there exists $x \in K_{i,q}$ for which $\|x - y^x\|_2 = \sqrt{w^*_\sigma}$, the upper bound $h^{(2)}_y$ is attained) by representing the Lipschitz bound as a function of y^x.

$$h_y(y^x) = \max_{x \in K_{i,q}} \left\{ \frac{1}{(2\pi)^{\frac{n}{2}} |\mathcal{W}|^{\frac{1}{2}} w^*_\sigma} \|x - y^x\|_2 e^{-\frac{\|x-y^x\|_2^2}{2w^*_\sigma}} \right\}. \tag{5.30}$$

Then,

$$\int_{\mathcal{Y}^x} |T_{y^x}(y^x|x) - T_{y^x}(y^x|\overline{x})| \, dy^x \leq \int_{\mathcal{Y}^x} h_y(y^x) \|x - \overline{x}\| \, dy^x \tag{5.31}$$

$$\begin{aligned}
\leq \delta^x_{i,q} & \int_{\{y^x : \|x-y^x\|_2^2 \leq w^*_\sigma, y^x \in \mathcal{Y}^x, x \in K_{i,q}\}} h^{(2)}_y \, dy^x \\
& + \delta^x_{i,q} \int_{\{y^x : \|x-y^x\|_2^2 > w^*_\sigma, y^x \in \mathcal{Y}^x, x \in K_{i,q}\}} h_y(y^x) \, dy^x \\
& = \delta^x_{i,q} \beta^y_{1,i} h^{(2)}_y \\
& + \delta^x_{i,q} \int_{\{y^x : \|x-y^x\|_2^2 > w^*_\sigma, y^x \in \mathcal{Y}^x, x \in K_{i,q}\}} h_y(y^x) \, dy^x
\end{aligned} \tag{5.32}$$

We use the change of variable $v = \|x^* - y^x\|_2$, with $x^* = \arg\min_{x \in K_{i,q}} \|x - y^x\|$, to rewrite the second term of (5.32), and apply an identity for integrals of polynomials.

$$\begin{aligned}\int_{\{y^x : \|x-y^x\|^2 > w_\sigma^*, y^x \in \overline{K}, x \in K_{i,q}\}} h_y(y^x)\, dy^x &= \frac{1}{(2\pi)^{\frac{n}{2}} |\mathcal{W}|^{\frac{1}{2}} w_\sigma^*} \int_{\sqrt{w_\sigma^*}}^{\infty} \frac{1}{2} v^2 e^{-\frac{v^2}{2w_\sigma^*}}\, dv \\ &\leq \frac{1}{(2\pi)^{\frac{n}{2}} |\mathcal{W}|^{\frac{1}{2}} w_\sigma^*} \int_0^{\infty} \frac{1}{2} v^2 e^{-\frac{v^2}{2w_\sigma^*}}\, dv \\ &\leq \frac{1}{(2\pi)^{\frac{n}{2}} |\mathcal{W}|^{\frac{1}{2}} w_\sigma^*} \frac{w_\sigma^*}{2} \sqrt{2 w_\sigma^* \pi} \qquad (5.33)\end{aligned}$$

Inserting (5.33) into (5.32) proves the lemma. ■

A similar result holds for the integral of T_x over $\mathcal{X}$.

Lemma 5.4. *For any $x, \overline{x} \in K_{i,q}$, for all $i = 1, \dots, m_q$, $q \in \mathcal{Q}$, the following holds:*

$$\int_{\mathcal{X}} |T_x(x'|q', x, q, u) - T_x(x'|q', \overline{x}, q, u)|\, dx \leq \left[\beta_{1,i}^x h_x^{(2)} + \beta_2^x\right] \delta_{i,q}^x$$

with $\beta_{1,i}^x = \int_{\{x' : \|x' - Ax - g(q,u,q')\|_2^2 \leq v_\sigma^*, x' \in \mathcal{X}, x \in K_{q,i}\}} 1\, dx$ and $\beta_2^x = \frac{a_\sigma^* \phi_v^*}{2} \sqrt{2 v_\sigma^* \pi}$, with a_σ^* the largest singular value of A.

The proof follows that of Lemma 5.3 with mean and covariance appropriate to T_x.

In order to show convergence of (5.26) to (5.4), we need some additional definitions. First, similarly to $\hat{\sigma}$, we define piecewise constant function $\hat{\alpha}$ as $\hat{\alpha}_n(s) = \alpha_{n,\delta}(\xi(s))$, so that

$$\hat{\alpha}_n(s) = \int_{\mathcal{S}} \int_{\mathcal{Y}} \hat{\alpha}_{n+1}^{*(y)}(s') \tau_y(dy|\xi(s'), u) \tau_s(ds'|\xi(s), u) 1_{K_\delta}(\xi(s)). \qquad (5.34)$$

We also define $\tilde{\alpha}_n(s)$ in the same manner as $\hat{\alpha}_n(s)$, except that it is directly related to $\alpha_n(s)$, i.e. uses the same optimal control input u, and the same combination of α_{n+1}-functions (determined by $*(y)$). In other words, $\tilde{\alpha}_n(s)$ is identical to $\alpha_n(s)$ in terms of the optimal policy tree from time n to N, but the values are calculated using $\tau_y(y|\xi(s'), u)$ and $\tau_s(s'|\xi(s), u)$.

$$\tilde{\alpha}_n^i(s) = \int_{\mathcal{S}} \int_{\mathcal{Y}} \tilde{\alpha}_{n+1}^{i(y)}(s') \tau_y(dy|\xi(s'), u^i) \tau_s(s'|\xi(s), u^i) 1_{K_\delta}(\xi(s)) \qquad (5.35)$$

for a specific α-function i associated with α_n^i. The superscript i for u^i and $i(y)$ indicates that the same choice of u and combination of $\alpha_{n+1}^j(s)$ are used for both $\alpha_n^i(s)$ and $\tilde{\alpha}_n^i(s)$. A bound on the difference between $\alpha_n^i(s)$ and $\tilde{\alpha}_n^i(s)$ is given in the following lemma.

Lemma 5.5. *For any $n \in [0, N]$, and any function $\alpha_n^i(s) \in \Gamma_n$, the associated function $\tilde{\alpha}_n^i(s)$ defined in (5.35) satisfies*

$$|\alpha_n^i(s) - \tilde{\alpha}_n^i(s)| \leq (N-n)N_q \left[\beta_1^y h_y^{(2)} + \beta_1^x h_x^{(2)} + \beta_2^y + \beta_2^x\right] \delta^x$$

for all $s \in \mathcal{S}$. The constants β_1^y and β_1^x are equal to $\max_{i=1,\ldots,m_q,q\in\mathcal{Q}} \beta_{1,i}^y$ and $\max_{i=1,\ldots,m_q,q\in\mathcal{Q}} \beta_{1,i}^x$ from Lemmas 5.3 and 5.4, respectively.

Proof. By induction. At time N,

$$|\alpha_N^i(s) - \tilde{\alpha}_N^i(s)| = \left|\int_{\mathcal{S}} \left(1_K(s) - 1_{K_\delta}(\xi(s))\right) ds\right| = 0 \tag{5.36}$$

since for any $s \in K$, by definition $\xi(s) \in K_\delta$. Assume for all $j = N-1, \ldots, n+1$, $|\alpha_j^i(s) - \tilde{\alpha}_j^i(s)| \leq (N-n)N_q \left[\beta_1^y h_y^{(2)} + \beta_1^x h_x^{(2)} + \beta_2^y + \beta_2^x\right] \delta^x$. For $j = n$,

$$\begin{aligned}
|\alpha_n^i(s) - \tilde{\alpha}_n^i(s)| = \bigg| &\int_{\mathcal{S}}\int_{\mathcal{Y}} \alpha_{n+1}^{i(y)}(s')\tau_y(dy|s', u^i)\tau_s(ds'|s, u^i)1_K(s) \\
&- \int_{\mathcal{S}}\int_{\mathcal{Y}} \tilde{\alpha}_{n+1}^{i(y)}(s')\tau_y(dy|\xi(s'), u^i)\tau_s(ds'|\xi(s), u^i)1_{K_\delta}(\xi(s))\bigg| \qquad (5.37) \\
\leq &\int_{\mathcal{S}}\int_{\mathcal{Y}} \left|\alpha_{n+1}^{i(y)}(s') - \tilde{\alpha}_{n+1}^{i(y)}(s')\right| \tau_y(dy|s', u^i)\tau_s(ds'|s, u^i)1_K(s) \\
&+ \int_{\mathcal{S}}\int_{\mathcal{Y}} \tilde{\alpha}_{n+1}^{i(y)}(s') \left|\tau_y(dy|s', u^i) - \tau_y(dy|\xi(s'), u^i)\right| \qquad (5.38) \\
&\qquad \times \tau_s(ds'|s, u^i)1_K(s) \\
&+ \int_{\mathcal{S}}\int_{\mathcal{Y}} \tilde{\alpha}_{n+1}^{i(y)}(s')\tau_y(dy|\xi(s'), u^i) \\
&\qquad \times \left|\tau_s(ds'|s, u^i) - \tau_s(ds'|\xi(s), u^i)\right| 1_K(s) \qquad (5.39) \\
\leq &\left|\alpha_{n+1}^{i(y)}(s') - \tilde{\alpha}_{n+1}^{i(y)}(s')\right| + N_q \left[\beta_1^y h_y^{(2)} + \beta_2^y\right] \delta^x \\
&\qquad + N_q \left[\beta_1^x h_x^{(2)} + \beta_2^x\right] \delta^x \qquad (5.40)
\end{aligned}$$

The second term of (5.39) simplifies according to Lemma 5.3 and noting that $\alpha(s)$ represents a probability that is bounded above by one. The third term simplifies according to Lemma 5.4. The term $1_K(s)$ does not affect the bound, and only indicates that both $\alpha_n(s)$ and $\tilde{\alpha}_n(s)$ are equal to zero for $s \notin K$. Applying the induction hypothesis to (5.40) gives the desired result. ■

We now can show convergence of the approximate viability probability over the discretized state space to the true viability probability.

Theorem 5.2. *For any time $n \in [0, N]$, and any $\sigma \in \Sigma$, $\sigma_\delta \in \Sigma_\delta$, the error between the value function* (5.11) *and the value function* (5.25) *based on the finite state approximation is bounded above by*

$$\left|V_n^*(\sigma_n) - V_{n,\delta}^*(\sigma_{n,\delta})\right| \leq \eta_n^\alpha \delta^x$$

with $\eta_n^\alpha = N_q \lambda \eta_n^\sigma + (N-n) N_q (\beta_1^y h_y^{(2)} + \beta_1^x h_x^{(2)} + \beta_2^y + \beta_2^x)$.

Specifically, the viability probability for PODTSHS $\mathcal{H}^{po}$ over time horizon N satisfies

$$\left|p_{\text{viab}}^N(\rho; K) - p_{\text{viab}}^N(\rho_\delta; K_\delta)\right| \leq \left[N_q N (\beta_1^y h_y^{(2)} + \beta_1^x h_x^{(2)} + \beta_2^y + \beta_2^x)\right] \delta^x.$$

Proof. By construction. At any time $n \in [0, N]$, given $\sigma_n \in \Sigma$ and $\sigma_{n,\delta} \in \Sigma_\delta$, we can rewrite the value function evaluated at σ in terms of α-functions.

$$\left|V_n^*(\sigma_n) - V_{n,\delta}^*(\sigma_{n,\delta})\right| = \left|\sup_{\alpha_n^i \in \Gamma_n} \langle \alpha_n^i, \sigma_n \rangle - \sup_{\alpha_{n,\delta}^i \in \Gamma_{n,\delta}} \langle \alpha_{n,\delta}^i, \sigma_{n,\delta} \rangle\right| \tag{5.41}$$

$$= \left|\langle \alpha_n^k, \sigma_n \rangle - \langle \alpha_{n,\delta}^l, \sigma_{n,\delta} \rangle\right| \tag{5.42}$$

Assume without loss of generality that $\langle \alpha_n^k, \sigma_n \rangle \geq \langle \alpha_{n,\delta}^l, \sigma_{n,\delta} \rangle$. Then, because

$\langle \tilde{\alpha}_n^k, \hat{\sigma}_n \rangle \leq \langle \hat{\alpha}_n^l, \hat{\sigma}_n \rangle$ by definition of the optimality of $\hat{\alpha}_n^l$, we can write

$$\begin{aligned} \left| V_n^*(\sigma_n) - V_{n,\delta}^*(\sigma_{n,\delta}) \right| &= \langle \alpha_n^k, \sigma_n \rangle - \langle \alpha_{n,\delta}^l, \sigma_{n,\delta} \rangle && (5.43) \\ &\leq \langle \alpha_n^k, \sigma_n \rangle - \langle \tilde{\alpha}_{n,\delta}^k, \sigma_{n,\delta} \rangle && (5.44) \\ &\leq \left| \langle \alpha_n^k, \sigma_n \rangle - \langle \tilde{\alpha}_n^k, \sigma_n \rangle \right| + \left| \langle \tilde{\alpha}_n^k, \sigma_n \rangle - \langle \tilde{\alpha}_n^k, \hat{\sigma}_n \rangle \right| && (5.45) \\ &\leq \int_{\mathcal{S}} \left| \alpha_n^k(s) - \tilde{\alpha}_n^k(s) \right| \sigma_n(ds) + \int_{\mathcal{S}} \tilde{\alpha}_n^k(s) \left| \sigma_n(ds) - \hat{\sigma}_n(ds) \right| \\ & && (5.46) \end{aligned}$$

Applying Lemma 5.5 to the first term of (5.46), and noting that the integral in the second term is in fact taken over K rather than $\mathcal{S}$ since $\tilde{\alpha}_n^k(s)$ is zero for all $s \notin K$, we obtain

$$\left| V_n^*(\sigma_n) - V_{n,\delta}^*(\sigma_{n,\delta}) \right| \leq (N-n) N_q \left[\beta_1^y h_y^{(2)} + \beta_1^x h_x^{(2)} + \beta_2^y + \beta_2^x \right] \delta^x + N_q \lambda \eta_n^\sigma \delta^x \quad (5.47)$$

which completes the proof. ■

Theorem 5.2 shows that the finite state approximation $\hat{\mathcal{H}}^{po}$ provides a means to approximately compute (5.4) through the viability probability for $\hat{\mathcal{H}}^{po}$, (5.26). As $\delta^x \to 0$, the finite state viability probability (5.26) converges to the true value (5.4), and the policy π_δ^* converges to π^*.

5.5.3 Gaussian Mixture Approximation

We now consider a different approximation by representing the information state σ and α-functions from Lemma 5.1 as Gaussian mixtures. That is, the information states and α-functions are each characterized by a set of weights, means, and covariances, dependent on the discrete mode q.

Difficulty arises from the incorporation of the indicator function 1_K in (5.14) and (5.10). Integration over the compact set K rather than all of $\mathcal{S}$ violates the preservation of the Gaussian form of σ under operator $\Phi_{y,u}$, and similarly for the α-functions. To preserve the Gaussian mixture structure, we therefore propose a radial basis function (RBF) approximation [PS91] to the indicator function, using Gaussians as the basis function. For each K_q, we set

$$1_{K_q}(x) \approx \sum_{i=1}^{I_q} w_i^I(q)\phi(x;\mu_i^I(q),\Sigma_i^I(q)) \tag{5.48}$$

using the most general form of the RBF with covariance Σ_i rather than ϵI. For simplicity we will denote $\phi(x;\mu_i^I(q),\Sigma_i^I(q))$ by $\phi_i^I(x)$. This approximation is valid since the RBFs are dense in L^p [PS91], i.e. given any function f in L^p, a weighted combination of RBFs can approximate f to arbitrary accuracy given enough components, and 1_K is in L^1.

However, the discontinuity in 1_{K_q} produces the Gibbs phenomenon at the boundary of K_q in the RBF approximation. Although these oscillations will always exist for a finite number of components, they could possibly be mitigated [FF11]. The oscillations can be constrained to a smaller region of K (shorter wavelength) with the addition of more components, indicating that the L^p error can be reduced but the pointwise error may not. Because we are interested only in integrating over K, this works to our advantage.

We define a new operator Φ^g and a new α-function $\alpha^g_{y,u,\sigma}$ by inserting the RBF approximation (5.48) into (5.10) and (5.14), respectively.

$$\left(\Phi^g_{y,u}\sigma_g\right)(s') = \frac{1}{\mathbb{P}(y|\sigma_g,u)}\tau_y(y|s',u) \times \sum_{q\in\mathcal{Q}}\int_{\mathcal{X}}\left[\sum_{i=1}^{I_q} w_i^I(q)\phi_i^I(x)\right]\tau_s(s'|s,u)\sigma_g(s)\,dx \tag{5.49}$$

$$\alpha^g_{y,u,\sigma}(s) = \sum_{q'\in\mathcal{Q}}\int_{\mathcal{X}}\alpha^{*(y)}_{n+1,g}(s') \times \tau_y(y|s',u)\tau_s(s'|s,u)\,dx'\left[\sum_{i=1}^{I_q} w_i^I(q)\phi_i^I(x)\right] \tag{5.50}$$

We presume continuous observations, as in Section 5.5.2 (the inclusion of a finite number of observations will be addressed in Section 5.6). We provide two lemmas stating that the operator $\Phi^g_{y,u}$ (5.49) and the α-function update (5.50) preserve the Gaussian mixture representation of $\sigma_{n,g}$ and $\alpha_{n,g}$ for all n.

Lemma 5.6. *The operator $\Phi^g_{y,u}$ (5.49) is closed under Gaussian mixtures, i.e. for σ_g a Gaussian mixture with L components, $\Phi^g_{y,u}\sigma_g$ is also a Gaussian mixture with N_qI_qL components for any $u \in \mathcal{U}$, $y \in \mathcal{Y}$.*

Lemma 5.7. *The expression (5.50) is closed under Gaussian mixtures, i.e. if $\alpha^{*(y)}_{n+1,g}$ is a Gaussian mixture with M components, $\alpha^g_{y,u,\sigma}$ is also a Gaussian mixture with N_qI_qM components, for any $u \in \mathcal{U}$, $y \in \mathcal{Y}$, $\sigma \in \Sigma$.*

The proofs of Lemmas 5.6 and 5.7 are straightforward and can be shown through extensive manipulation of products of Gaussian pdfs. Lemma 5.6 implies that we can approximate σ through a Gaussian mixture and use the equivalent update operator $\Phi^g_{y,u}$, hence the Gaussian mixture approximation of σ is

$$\begin{aligned} \sigma_{0,g}(x,q) &= \sigma_0(x,q) = Q_0(q)\phi(x;\mu_0;\Sigma_0) \\ \sigma_{n,g}(x,q) &= \sum_{l=1}^{L} w^\sigma_{l,n}(q)\phi(x;\mu^\sigma_{l,n}(q),\Sigma^\sigma_{l,n}(q)) \end{aligned} . \tag{5.51}$$

Similarly, the Gaussian mixture approximation of any α-function is written:

$$\begin{aligned} \alpha_{N,g}(x,q) &= \sum_{i=1}^{I_q} w^I_i(q)\phi^I_i(x) \\ \alpha_{n,g}(x,q) &= \sum_{m=1}^{M} w^\alpha_{m,n}(q)\phi(x;\mu^\alpha_{m,n}(q),\Sigma^\alpha_{m,n}(q)) \end{aligned} . \tag{5.52}$$

The weights, means, and covariances are defined recursively. Their exact representations are lengthy, but again easy to derive through manipulations of Gaussians. Appendix B provides a proof of Lemma 5.6 and Lemma 5.7, and gives the recursive expressions for $\alpha_{n,g}(x,q)$ and $\sigma_{n,g}(x,q)$.

Note that although the Gaussian mixture representation of $\alpha^g_{y,u,\sigma}$ has a finite number of components given that the representation of $\alpha_{n+1,g}$ is finite, the actual α-function, $\alpha_{n,g}$, is expressed as the integral of $\alpha^g_{y,u,\sigma}$ over $\mathcal{Y}$. Therefore, without the assumption that $\mathcal{Y}$ is finite, $\alpha_{n,g}$ must have an infinite number of components (by breaking the integral over $\mathcal{Y}$ into a summation over regions of size $\Delta \subset \mathcal{Y}$ and taking the limit as $\Delta \to 0$). We take some liberty in overlooking this discrepancy, because it does not affect the proofs in this section. We impose a finite set $\mathcal{Y}$ in Section 5.6, which makes the Gaussian mixture representation of the α-functions indeed valid, and discuss additional error implications.

The viability problem for the Gaussian mixture approximation is defined as

$$\begin{aligned}\sup_{\pi_g \in \Pi_g} p^N_{\text{viab}}(\pi_g, \rho_g; K_g) &= \sup_{\pi_g \in \Pi_g} \mathbb{P}^{\pi_g}\left[s_n \in K_g, \ \forall\, n \in [0, N]\right] \\ &= V^*_{0,g}(\rho_g) \end{aligned} \tag{5.53}$$

with K_g an approximation of K according to (5.48). The value function $V^*_{n,g}(\sigma_{n,g})$ is described through the recursion

$$\begin{aligned} V^*_{N,g}(\sigma_g) &= \sum_{\mathcal{Q}} \int_{\mathcal{X}} \sum_{i=1}^{I_q} w^I_i(q) \phi^I_i(x) \sigma_g(x, q)\, dx \\ V^*_{n,g}(\sigma_g) &= \max_{u \in \mathcal{U}} \int_{\mathcal{Y}} V^*_{n+1,g}(\Phi^g_{y,u}\sigma_g) \mathbb{P}(dy|\sigma_g, u) \end{aligned} . \tag{5.54}$$

Since T_x, T_{y^x}, and ρ are Gaussian, the Gaussian mixture representation of α and σ are exact, aside from the approximation of 1_K using RBFs. To quantify the error incurred from calculating $V^*_{0,g}$ as opposed to V^*_0 (from integration of (5.48) over $\mathcal{S}$ rather than over K), we define the error

$$\epsilon_I = \left\| 1_K - \sum_{i=1}^{I_q} w^I_i(q) \phi^I_i(x) \right\|_{L^1(\mathcal{S})} . \tag{5.55}$$

We additionally constrain the RBF approximation (5.48). The weights $w^I_i(q)$ must satisfy the following three conditions.

$$\begin{aligned} \int_{\mathcal{X}} \sum_{i=1}^{I_q} w^I_i(q) \phi^I_i(x)\, dx &\leq \int_{K_q} 1\, dx \,\forall\, q \in \mathcal{Q} \\ \int_{\mathcal{S}} \sum_{i=1}^{I_q} w^I_i(q') \phi^I_i(x') \tau_s(s'|s, u)\, ds' &\leq 1 \,\forall\, s \in \mathcal{S}, s' \in \mathcal{S}, u \in \mathcal{U} \\ \int_{\mathcal{S}} \sum_{i=1}^{I_q} w^I_i(q) \phi^I_i(x) \tau_s(s'|s, u)\, ds &\leq 1 \,\forall\, s \in \mathcal{S}, s' \in \mathcal{S}, u \in \mathcal{U} \end{aligned} \tag{5.56}$$

The first condition assures that $\int_{\mathcal{S}} \sum_{i=1}^{I_q} w_i^I(q)\phi_i^I(x)\,ds \leq N_q\lambda$. The second and third conditions assure that no probability exceeds one. All conditions are easily satisfied by calculating the optimal weights and reducing them slightly if necessary.

Information State Approximation Error

The error between σ and σ_g is stated in terms of the L^1 norm on $\mathcal{S}$, although a nearly identical result is available for the pointwise error.

Theorem 5.3. *The Gaussian mixture approximation $\sigma_{n,g}$ of σ satisfies*

$$\left\|\sigma_n - \sigma_{n,g}\right\|_{L^1(\mathcal{S})} \leq \gamma_n^\sigma \epsilon_I$$

for any $n \in [0,N]$, $y \in \mathcal{Y}$, and $u \in \mathcal{U}$, with $\gamma_n^\sigma = \sum_{j=1}^{n} (\phi_w^*)^j \phi_{\sigma,j}^*$ and $\phi_{\sigma,j}^* = \max_{l \in 1,\ldots,L; q \in \mathcal{Q}} (2\pi)^{-\frac{n}{2}} |\Sigma_{l,j}^\sigma(q)|^{-\frac{1}{2}}$.

Proof. By induction. At time zero, $\sigma_{0,g}(s) = \sigma_0(s)$, so that $\|\sigma_0 - \sigma_{0,g}\|_1 = 0$. Assume that $\|\sigma_i - \sigma_{i,g}\|_{L^1(\mathcal{S})} \leq \gamma_i^\sigma \epsilon_I$ for all $i = 1,\ldots,n$. Then at time $n+1$ we have, for some $y \in \mathcal{Y}$ and $u \in \mathcal{U}$,

$$\begin{aligned} \left\|\sigma_{n+1} - \sigma_{n+1,g}\right\|_{L^1(\mathcal{S})} \leq & \int_{\mathcal{S}} \tau_y(y|s',u) \int_{\mathcal{S}} \Bigg| 1_K(s)\sigma_n(ds) \\ & \qquad - \sum_{i=1}^{I_q} w_i^I(q)\phi_i^I(x)\sigma_{n,g}(ds) \Bigg| \tau_s(ds'|s,u) \\ \leq & \; \phi_w^* \Bigg[\int_{\mathcal{S}} \left|1_K(s)\sigma_n(ds) - 1_K(s)\sigma_{n,g}(ds)\right| \\ & \qquad + \int_{\mathcal{S}} \Bigg| 1_K(s)\sigma_{n,g}(ds) - \sum_{i=1}^{I_q} w_i^I(q)\phi_i^I(x)\sigma_{n,g}(ds) \Bigg| \Bigg] \end{aligned} \tag{5.57}$$

$$\left\|\sigma_{n+1}-\sigma_{n+1,g}\right\|_{L^1(\mathcal{S})} \leq \phi_w^* \Bigg[\left\|\sigma_n-\sigma_{n,g}\right\|_1 + \int_{\mathcal{S}} \left| 1_K(s) - \sum_{i=1}^{I_q} w_i^I(q)\phi_i^I(x) \right| \sigma_{n,g}(ds) \Bigg] \tag{5.58}$$

The first term of line (5.58) follows because the integral over K is less than the integral over all of $\mathcal{S}$, since K is a compact subset of $\mathcal{S}$. The induction hypothesis completes the proof. ■

Convergence of (5.53) to (5.4)

As for the proof of Theorem 5.2, we define the function $\tilde{\alpha}_{n,g}^i(s)$ which utilizes the same policy tree as $\alpha_n^i(s)$ for a specific $\alpha_n^i(s) \in \Gamma_n$.

$$\tilde{\alpha}_{n,g}^i(s) = \int_{\mathcal{S}}\int_{\mathcal{Y}} \tilde{\alpha}_{n+1,g}^{i(y)}(s')\tau_y(dy|s',u^i)\tau_s(ds'|s,u^i)ds' \left[\sum_{i=1}^{I_q} w_i^I(q)\phi_i^I(x)\right] \tag{5.59}$$

with u^i the optimal control input associated with $\alpha_n^i(s)$ and $i(y)$ indicating that $\alpha_{n+1,g}^{i(y)}(s)$ is chosen according to the indices selected by $*(y)$ for $\alpha_n(s)$. The following lemma describes the relation between $\alpha_n(s)$ and $\tilde{\alpha}_n(s)$.

Lemma 5.8. *For any $n \in [0,N]$, and any $\alpha_n^i(s) \in \Gamma_n$, the associated function $\tilde{\alpha}_{n,g}^i(s)$ defined in (5.59) satisfies*

$$\left\|\alpha_n^i(s) - \tilde{\alpha}_{n,g}^i\right\|_{L^1(\mathcal{S})} \leq \left(\sum_{k=n}^{N}(\lambda\phi_v^*)^{N-k}\right)\epsilon_I$$

Proof. By induction. At time N,

$$\left\|\alpha_N(s)-\tilde{\alpha}_{N,g}\right\|_{L^1(\mathcal{S})} = \int_{\mathcal{S}} \left| 1_K(s) - \sum_{i=1}^{I_q} w_i^I(q)\phi_i^I(x)\right| ds \tag{5.60}$$

$$= \epsilon_I \tag{5.61}$$

and the result is satisfied. Assume for $j = N-1,\ldots,n+1$ that $\|\alpha_j^i(s) - \tilde{\alpha}_{j,g}^i(s)\|_1 \leq$

$\left(\sum_{k=j}^{N}(\lambda\phi_v^*)^{N-k}\right)\epsilon_I$ for any $\alpha_j^i \in \Gamma_j$. Then for $j = n$,

$$\left\|\alpha_n^i(s) - \tilde{\alpha}_{n,g}^i(s)\right\|_1 = \int_{\mathcal{S}}\left|\int_{\mathcal{S}}\int_{\mathcal{Y}} \alpha_{n+1}^{i(y)}(s')\tau_y(dy|s',u^i)\tau_s(ds'|s,u^i)1_K(s)\right.$$
$$\left. - \tilde{\alpha}_{n+1,g}^{i(y)}(s')\tau_y(dy|s',u^i)\tau_s(ds'|s,u^i)\sum_{i=1}^{I_q} w_i^I(q)\phi_i^I(x)\right| ds \tag{5.62}$$

$$\leq \int_{\mathcal{S}}\int_{\mathcal{S}}\int_{\mathcal{Y}}\left|\alpha_{n+1}^{i(y)}(s') - \tilde{\alpha}_{n+1,g}^{i(y)}(s')\right|\tau_y(dy|s',u^i)$$
$$\times\, \tau_s(ds'|s,u^i)1_K(s)ds$$
$$+ \int_{\mathcal{S}}\int_{\mathcal{S}}\int_{\mathcal{Y}}\tilde{\alpha}_{n+1,g}^{i(y)}(s')\tau_y(dy|s',u^i)$$
$$\times\, \tau_s(ds'|s,u^i)\left|1_K(s) - \sum_{i=1}^{I_q} w_i^I(q)\phi_i^I(x)\right| ds \tag{5.63}$$

$$\leq \int_{\mathcal{S}}\left|\alpha_{n+1}^{i(y)}(s') - \tilde{\alpha}_{n+1,g}^{i(y)}(s')\right|\phi_v^* ds' \int_{\mathcal{S}} 1_K(s)ds$$
$$+ \int_{\mathcal{S}}\left|1_K(s) - \sum_{i=1}^{I_q} w_i^I(q)\phi_i^I(x)\right| ds \tag{5.64}$$

$$\leq \lambda\phi_v^*\left(\sum_{k=n+1}^{N}(\lambda\phi_v^*)^{N-k}\right)\epsilon_I + \epsilon_I \tag{5.65}$$

$$= \left(\sum_{k=n}^{N}(\lambda\phi_v^*)^{N-k}\right)\epsilon_I \tag{5.66}$$

■

Theorem 5.4. *For any time $n \in [0, N]$, and any $\sigma \in \Sigma$, $\sigma_g \in \Sigma_g$, the error between the value function (5.11) given σ and the value function (5.54) given σ_g using the Gaussian mixture approximation is bounded above by*

$$\left|V_n^*(\sigma_n) - V_{n,g}^*(\sigma_{n,g})\right| \leq \gamma_n^\alpha \epsilon_I$$

with $\gamma_n^\alpha = \left(\sum_{k=n}^{N}(\lambda\phi_v^*)^{N-k}\right)\phi_{\sigma,n}^* + \gamma_n^\sigma$.

Specifically, the viability probability for PODTSHS $\mathcal{H}^{po}$ over time horizon N satisfies

$$\left|p^N_{\text{viab}}(\rho; K) - p^N_{\text{viab}}(\rho_g; K_g)\right| \leq \left[\sum_{k=0}^{N} (\lambda\phi^*_v)^{N-k}\, \phi^*_{\sigma,0}\right] \epsilon_I.$$

Proof. By construction. For any time $n \in [0, N]$, given $\sigma_n \in \Sigma$ and $\sigma_{n,g} \in \Sigma_g$, we can rewrite the value function evaluated at σ in terms of α-functions.

$$\left|V^*_n(\sigma_n) - V^*_{n,g}(\sigma_{n,g})\right| = \left|\sup_{\alpha^i_n \in \Gamma_n} \langle \alpha^i_n, \sigma_n\rangle - \sup_{\alpha^i_{n,g} \in \Gamma_{n,g}} \langle \alpha^i_{n,g}, \sigma_{n,g}\rangle\right| \tag{5.67}$$

$$= \left|\langle \alpha^k_n, \sigma_n\rangle - \langle \alpha^l_{n,g}, \sigma_{n,g}\rangle\right| \tag{5.68}$$

As in the proof of Theorem 5.2, assume without loss of generality that $\langle \alpha^k_n, \sigma_n\rangle \geq \langle \alpha^l_{n,g}, \sigma_{n,g}\rangle$.

$$\left|V^*_n(\sigma_n) - V^*_{n,g}(\sigma_{n,g})\right| = \langle \alpha^k_n, \sigma_n\rangle - \langle \alpha^l_{n,g}, \sigma_{n,g}\rangle \tag{5.69}$$

$$\leq \langle \alpha^k_n, \sigma_n\rangle - \langle \tilde{\alpha}^k_{n,g}, \sigma_{n,g}\rangle \tag{5.70}$$

$$\leq \left|\langle \alpha^k_n, \sigma_n\rangle - \langle \tilde{\alpha}^k_{n,g}, \sigma_n\rangle\right| + \left|\langle \tilde{\alpha}^k_{n,g}, \sigma_n\rangle - \langle \tilde{\alpha}^k_{n,g}, \sigma_{n,g}\rangle\right| \tag{5.71}$$

$$\leq \int_{\mathcal{S}} \left|\alpha^k_n(s) - \tilde{\alpha}^k_{n,g}(s)\right| \sigma_n(ds) + \int_{\mathcal{S}} \tilde{\alpha}^k_{n,g}(s) \left|\sigma_n(ds) - \sigma_{n,g}(ds)\right| \tag{5.72}$$

$$\leq \left(\sum_{k=n}^{N} (\lambda\phi^*_v)^{N-k}\right) \phi^*_{\sigma,n}\epsilon_I + \gamma^\sigma_n \epsilon_I \tag{5.73}$$

■

Theorem 5.4 shows that the convergence of the Gaussian mixture approximation of both σ and the value function depends on the L^1 error between the indicator function over K and the RBF approximation (5.48), rather than the pointwise error. Although the pointwise error may not converge to zero for a finite number of components in the RBF, the integral of the error can be small, as we will show in Section 5.8.

5.6 Approximate Numerical Solution with Lower Bound

A numerical solution of Problem 5.1 via either a discrete or Gaussian mixture approximation additionally requires sets Γ_n and Σ to be finite, whereas we have sets of infinite size because of the uncountable nature of $\mathcal{Y}$. However, a lower bound on the viability probabilities (5.26) and (5.53) can still be obtained, by characterizing the error that results from using $\tilde{\Gamma}_n \subset \Gamma_n$ and $\tilde{\Sigma} \subset \Sigma$, finite collections of α-functions and information states, respectively.

We again exploit point-based approximation methods described in Section 5.4. We examine the generation of subsets of the information states and α-functions, and prove that each guarantees a lower bound to the viability probability of whichever approximation of Section 5.5 we choose. In contrast to most point-based solvers, we do not assume a finite set of observations, and hence discretize the observations merely for the computation of the α-functions. Combining belief space sampling with discretized observations assures a lower bound to the viability probability.

5.6.1 Sampling from the information space

We characterize the error from using a sampled subset of Σ for performing backup operations (as in (5.7)). Presume that a finite number of information states has been generated according to one of the many methods available [SPK13]. We generate a finite set $\tilde{\Gamma}_n$ of α-functions, one for each $\sigma \in \tilde{\Sigma}$. The convexity of the value functions guarantees that the subset $\tilde{\Gamma}_n$ provides a lower bound on V_n^*. Further, we can show that the error between the approximate value function $\tilde{V}_n^*$ and the true value function V_n^* depends on how densely we sample Σ.

The value function $\tilde{V}_n^*$ is formally defined as $\tilde{V}_n^*(\sigma) = \sup_{\tilde{\alpha}_n \in \tilde{\Gamma}_n} \langle \tilde{\alpha}_n, \sigma \rangle$ with

$$\begin{aligned} \tilde{\alpha}_n(s) &= \int_{\mathcal{Y}} \int_{\mathcal{S}} \tilde{\alpha}_{n+1}^{*(y)}(s') \tau_y(dy|s', u) \tau_s(ds'|s, u) 1_K(s) \\ \tilde{\alpha}_{n+1}^{*(y)}(s') &= \arg \left\{ \sup_{\tilde{\alpha}_{n+1}^i \in \tilde{\Gamma}_{n+1}} \int_{\mathcal{S}} \tilde{\alpha}_{n+1}^i(s') \tau_y(y|s', u) \int_K \tau_s(ds'|s, u) \sigma(ds) \right\} \end{aligned} \tag{5.74}$$

so that $\tilde{V}_n^*$ is characterized by the finite set $\tilde{\Gamma}_n$ at each time step. We also define an intermediate value function $\hat{V}_n^* = \sup_{\hat{\alpha}_n \in \tilde{\Gamma}_n} \langle \hat{\alpha}_n, \sigma \rangle$ that generates $\tilde{\Gamma}_n$ recursively from Γ_{n+1}, i.e. that introduces one point-based backup from the full set Γ_{n+1}. Then $\hat{\alpha}_n$ is written as a function of α_{n+1}^* rather than $\tilde{\alpha}_{n+1}^*$, with

$$\alpha_{n+1}^{*(y)}(s') = \arg \left\{ \sup_{\alpha_{n+1} \in \Gamma_{n+1}} \int_{\mathcal{S}} \alpha_{n+1}(s') \tau_y(y|s', u) \int_K \tau_s(ds'|s, u) \sigma(ds) \right\}. \tag{5.75}$$

Finally, let δ^σ be the maximum L^1 distance between points in $\tilde{\Sigma}$ and points in Σ.

$$\delta^\sigma = \sup_{\tilde{\sigma} \in \tilde{\Sigma}} \inf_{\sigma \in \Sigma} \|\tilde{\sigma} - \sigma\|_1 \tag{5.76}$$

In the following, we do not distinguish between the vector and Gaussian mixture representations of σ and α, because the results apply to both cases.

Lemma 5.9. *For any $n \in [0, N]$ and $\sigma \in \Sigma$, the error introduced in one iteration of point-based value iteration is at most δ^σ.*

$$|\hat{V}_n^*(\sigma) - V_n^*(\sigma)| \leq \delta^\sigma$$

Proof. The proof is modified from [PGT06]. Let $\tilde{\sigma} \in \tilde{\Sigma}$ be the closest point in the L^1 sense to σ. Let $\hat{\alpha}^* \in \tilde{\Gamma}_n$ be maximal at $\tilde{\sigma}$, and $\alpha^* \in \Gamma_n$ (and not in $\tilde{\Gamma}_n$) is the function that *would* be maximal at σ had it been calculated.

Then

$$\begin{aligned} |\hat{V}_n^*(\sigma) - V_n^*(\sigma)| &\leq |\langle \alpha^*, \sigma \rangle - \langle \hat{\alpha}^*, \sigma \rangle| \\ &\leq |\langle \alpha^*, \sigma \rangle - \langle \hat{\alpha}^*, \sigma \rangle + \langle \alpha^*, \tilde{\sigma} \rangle - \langle \alpha^*, \tilde{\sigma} \rangle| \\ &\leq |\langle \alpha^*, \sigma \rangle - \langle \hat{\alpha}^*, \sigma \rangle + \langle \hat{\alpha}^*, \tilde{\sigma} \rangle - \langle \alpha^*, \tilde{\sigma} \rangle| \qquad (5.77) \\ &\leq |\langle \alpha^* - \hat{\alpha}^*, \sigma - \tilde{\sigma} \rangle| \\ &\leq \|\alpha^* - \hat{\alpha}^*\|_\infty \|\sigma - \tilde{\sigma}\|_1 \qquad (5.78) \\ &\leq \|\alpha^* - \hat{\alpha}^*\|_\infty \delta^\sigma \qquad (5.79) \end{aligned}$$

Line (5.77) follows because $\alpha^{(2)}$ is optimal for $\sigma^{(2)}$, implying $\langle \alpha^{(1)}, \sigma^{(2)} \rangle \leq \langle \alpha^{(2)}, \sigma^{(2)} \rangle$. Line (5.78) follows from Hölder's Inequality. Line (5.79) can be further simplified by noting that the α-functions are bounded between 0 and 1 for all $s \in \mathcal{S}$, and so $\|\alpha^{(1)} - \alpha^{(2)}\|_\infty \leq 1$.

■

We now use Lemma 5.9 to derive a bound between the true value function and the point-based approximation at any time n.

Theorem 5.5. *For a set of information states Σ, sampled set $\tilde{\Sigma}$, and any time $n \in [0, N]$ and any $\sigma \in \Sigma$, the error from using point-based value iteration versus full value iteration is bounded above by*

$$|\tilde{V}_n^*(\sigma) - V_n^*(\sigma)| \leq (N-n)\delta^\sigma.$$

Proof. By induction. At time N, $V_N^* = \tilde{V}_N^*$ for all σ and the inequality is trivially satisfied. Assume for all $i = n+1, \ldots, N-1$ that $|\tilde{V}_i^*(\sigma) - V_i^*(\sigma)| \leq (N-i)\delta^\sigma$. At time n,

$$\begin{aligned}
|\tilde{V}_n^*(\sigma) - V_n^*(\sigma)| &= |\tilde{V}_n^*(\sigma) - V_n^*(\sigma) + \hat{V}_n^*(\sigma) - \hat{V}_n^*(\sigma)| \\
&\leq |\tilde{V}_n^*(\sigma) - \hat{V}_n^*(\sigma)| + |\hat{V}_n^*(\sigma) - V_n^*(\sigma)| \\
&\leq \left| \int_{\mathcal{Y}} \sup_{\tilde{\Gamma}_{n+1}} \langle \tilde{\alpha}_{n+1}, \Phi_{y,u}\tilde{\sigma} \rangle \mathbb{P}(dy|\sigma, u) \right. \\
&\quad \left. - \int_{\mathcal{Y}} \sup_{\Gamma_{n+1}} \langle \alpha_{n+1}, \Phi_{y,u}\tilde{\sigma} \rangle \mathbb{P}(dy|\sigma, u) \right| + \delta^\sigma && (5.80) \\
&\leq |\tilde{V}_{n+1}^*(\sigma_{n+1}) - V_{n+1}^*(\sigma_{n+1})| + \delta^\sigma && (5.81) \\
&\leq (N-n-1)\delta^\sigma + \delta^\sigma && (5.82)
\end{aligned}$$

Line (5.80) follows from Lemma 5.9 and because $\hat{V}_n^*$ and $\tilde{V}_n^*$ are computed over the same set $\tilde{\Sigma}$ (allowing us to write $\hat{V}$ and $\tilde{V}$ in terms of $\tilde{\sigma} \in \tilde{\Sigma}$ corresponding to $\tilde{\alpha}_n^* \in \tilde{\Gamma}_n$), and line (5.82) from the induction hypothesis. ■

Thus the error between the point-based approximation and the actual value function is directly proportional to how densely $\tilde{\Sigma}$ is sampled, and converges to zero as $\tilde{\Sigma}$ approaches Σ.

5.6.2 Calculating the Alpha-functions

Over the uncountably infinite space $\mathcal{Y}$, we cannot calculate $\alpha_{y,u,\sigma}$ for all $y \in \mathcal{Y}$, despite a finite set of u and σ. We therefore compute a subset of $\alpha_{y,u,\sigma}$ for the finite set y^i, to approximate α_n as $\alpha_n(s) \approx \sum_{y^i} \alpha_{y^i,u,\sigma}(s)$. We discretize $\mathcal{Y}$ in a similar fashion to the discretization of $\mathcal{S}$ in Section 5.5.2.

However, since $\mathcal{Y}^x$ is not compact, we consider an expanded set $\overline{K} = \bigcup_{y^q \in \mathcal{Q}} \overline{K}_{y^q} \supset K$ defined so that the probability of observing a value y for $s \in K$ that is outside of $\overline{K}$ is approximately zero, i.e. $\tau_y(\mathcal{Y} \backslash \overline{K} | s \in K, u) < \epsilon$, $\epsilon \ll 1$. For example, $\overline{K}_{y^q} = \{x + \hat{w} : x \in K_{y^q}, ||\hat{w}|| \leq 3w^*\}$ with w^* the largest diagonal entry of $\mathcal{W}$ results in the probability of observing any y outside of $\overline{K}$ as less than 0.003, which can be essentially dismissed with minimal impact on resulting calculations. The sets $\overline{K}_{y^q}$ are divided into disjoint subsets $\overline{K}_{i,y^q}$, $\bigcup_{i=1,\dots l_q} \overline{K}_{i,y^q} = \overline{K}_{y^q}$. We also define $\psi_y = \overline{K}^c = \mathcal{Y} \backslash \overline{K}$, such that $\bigcup_{i=1,\dots,l_q} \overline{K}_{i,y^q} \times \{\psi_y\} = \mathbb{R}^n$.

The partition of $\overline{K}$ is denoted $\mathcal{G}^y = \bigcup_{i,y^q} \mathcal{G}^y_{i,y^q}$ with $\mathcal{G}^y_{i,y^q} = \{\overline{K}_{i,y^q} \times y^q : i = 1, \dots l_q, y^q \in \mathcal{Q}\}$. The diameter of partition $\overline{K}_{i,y^q}$ is $\delta^y_{i,y^q} = \sup\{\|y - \overline{y}\| : y, \overline{y} \in \overline{K}_{i,y^q}\}$, with maximum diameter $\delta^y = \max_{i,y^q} \delta^y_{i,y^q}$. Each partition $\mathcal{G}^y_{i,y^q}$ has a representative element (y^{x,i,y^q}, y^q) and a set $\mathcal{Y}_\delta = \{(y^{x,i,y^q}, y^q) : i = 1, \dots, l_q, y^q \in \mathcal{Q}\}$. The function $\theta : \mathcal{Y} \to \mathcal{Y}_\delta$ maps observation $y \in \mathcal{Y}$ to its representative value (y^{x,i,y^q}, y^q); the set-valued function $\Theta : \mathcal{Y}_\delta \to \overline{K}$ maps the point (y^{x,i,y^q}, y^q) to the corresponding set $\mathcal{G}_{i,y^q}$.

The finite observation space is $W_\delta = \mathcal{Y}_\delta \bigcup \{\psi_y\}$. For the finite state approximation, the transition function $\tau^\delta_y : W_\delta \times K_\delta \times \mathcal{U} \to [0, 1]$ is defined as

$$\tau^\delta_y(w|z, u) = \begin{cases} \tau_y(\Theta(w)|z, u), & \text{if } w \in \mathcal{Y}_\delta \\ 1 - \sum_{\overline{w} \in \mathcal{Y}_\delta} \tau_y(\Theta(\overline{w})|z, u), & \text{if } w = \psi_y \end{cases}. \tag{5.83}$$

For the Gaussian mixture approximation, we define the transition function τ_y^g in the same fashion as (5.83), but with

$$\tau_y(\Theta(w)|z,u) \approx T_{y^q}(y^q|q,u) \sum_{j=1}^{M_y} c_j \phi_j^y(y_j^{x,i,y^q}; x, \mathcal{W}) \tag{5.84}$$

so that the α-functions will also be Gaussian mixtures at each time step. This is further discussed in Appendex B. Note that $w = (y^{x,i,y^q}, y^q)$, y_j^{x,i,y^q} is a set of mesh points inside $\mathcal{G}_{i,y^q}$ associated with w, and c_j is a weight proportional to the mesh spacing (determined, e.g., by the trapezoidal rule for numerical integration).

Discretized observations for finite state approximation

We use $\tilde{\Gamma}_{n,\delta}$ and $\tilde{V}^*_{n,\delta}$ to denote the approximation using a finite subset of $\Gamma_{n,\delta}$, with the important distinction that the subset is now generated by a finite collection of observations (as opposed to $\tilde{\Sigma}$).

The value function is then

$$\tilde{V}^*_{n,\delta} = \sup_{\tilde{\alpha}_{n,\delta} \in \tilde{\Gamma}_{n,\delta}} \sum_{z \in K_\delta} \tilde{\alpha}_{n,\delta}(z)\sigma_\delta(z),$$

with

$$\tilde{\alpha}_{n,\delta}(z) = \sum_{w \in W_\delta} \tilde{\alpha}^\delta_{w,u,\sigma}(z) \tag{5.85}$$

$$\tilde{\alpha}^\delta_{w,u,\sigma}(z) = \sum_{z' \in K_\delta} \alpha^{*(w)}_{n+1,\delta}(z')\tau^\delta_y(w|z',u)\tau^\delta_s(z'|z,u)\sigma_\delta(z) \tag{5.86}$$

$$\tilde{\alpha}^{*(w)}_{n+1,\delta}(z') = \arg\left\{ \sup_{\tilde{\alpha}^i_{n+1,\delta} \in \tilde{\Gamma}_{n+1,\delta}} \sum_{z' \in K_\delta} \tilde{\alpha}^i_{n+1,\delta}(z')\tau^\delta_y(w|z',u)\tau^\delta_s(z'|z,u)\sigma_\delta(z) \right\}. \tag{5.87}$$

Similarly to (5.75), $\hat{V}^*_{n,\delta}(\sigma_\delta) = \sup_{\hat{\alpha} \in \tilde{\Gamma}_{n,\delta}} \sum_z \hat{\alpha}_{n,\delta}(z)\sigma_\delta(z)$ is the intermediate value function, with $\hat{\alpha}_{n,\delta}$ calculated using $\alpha^{*(w)}_{n+1,\delta} \in \Gamma_{n+1,\delta}$ (as opposed to $\tilde{\Gamma}_{n+1}$).

We can bound the error introduced in one iteration of approximating the α-vectors through discretized observations.

Lemma 5.10. *For any time $n \in [0, N]$ and $\sigma_\delta \in \Sigma_\delta$, the error between $V^*_{n,\delta}(\sigma_\delta)$ and $\hat{V}^*_{n,\delta}(\sigma_\delta)$ satisfies*

$$0 \leq V^*_{n,\delta}(\sigma_\delta) - \hat{V}^*_{n,\delta}(\sigma_\delta) \leq N_q \overline{\lambda} h_y^{(1)} \delta^y + \frac{\epsilon}{N}$$

given that the discretized observations w are chosen so that

$$\sum_{z' \in K_\delta} \alpha^{*(w)}_{n+1,\delta}(z') \tau^\delta_y(w|z', u) > \sum_{z' \in K_\delta} \alpha^{*(w)}_{n+1,\delta}(z') \tau_y(w|z', u) |\Theta(w)|,$$

and with $\overline{\lambda}$ the largest Lebesgue measure of sets $\overline{K}_{y^q}$.

Proof. Define $\overline{K}$ such that $\tau_y(\mathcal{Y} \backslash \overline{K} | z \in K_\delta, u) < \frac{\epsilon}{N}$. Then

$$\begin{aligned}
|V^*_{n,\delta}(\sigma_\delta) - \hat{V}^*_{n,\delta}(\sigma_\delta)| &= \left| \sup_{\alpha_{n,\delta} \in \Gamma_{n,\delta}} \sum_{z \in K_\delta} \alpha_{n,\delta}(z), \sigma_\delta(z) - \sup_{\hat{\alpha} \in \tilde{\Gamma}_{n,\delta}} \sum_z \hat{\alpha}_{n,\delta}(z) \sigma_\delta(z) \right| \\
&\leq \int_{\overline{K}} \sum_{z, z' \in K_\delta} \Big[\alpha^{*(y)}_{n+1,\delta}(z') \tau_y(dy|z', u^*) \tau^\delta_s(z'|z, u^*) \sigma_\delta(z) \\
&\qquad - \alpha^{*(\theta(y))}_{n+1,\delta}(z') \tau_y(dy|z', u^*) \tau^\delta_s(z'|z, u^*) \sigma_\delta(z) \Big] \\
&\quad + \int_{\mathcal{Y} \backslash \overline{K}} \sum_{z, z' \in K_\delta} \alpha^{*(y)}_{n+1,\delta}(z') \tau_y(dy|z', u^*) \tau^\delta_s(z'|z, u^*) \sigma_\delta(z) \\
&\quad - \int_{\mathcal{Y} \backslash \overline{K}} \sum_{z, z' \in K_\delta} \alpha^{*(\psi_y)}_{n+1,\delta}(z') \tau_y(dy|z', u^*) \tau^\delta_s(z'|z, u^*) \sigma_\delta(z) \\
&\leq \int_{\overline{K}} \sum_{z, z' \in K_\delta} \Big[\alpha^{*(y)}_{n+1,\delta}(z') \tau_y(dy|z', u^*) \tau^\delta_s(z'|z, u^*) \sigma_\delta(z) \\
&\quad - \alpha^{*(\theta(y))}_{n+1,\delta}(z') \tau_y(dy|z', u) \tau^\delta_s(z'|z, u^*) \sigma_\delta(z) \Big] + \frac{\epsilon}{N} \qquad (5.88)
\end{aligned}$$

Note that (5.88) is nonnegative, meaning that using $\hat{\alpha}^*_\delta$ produces a lower bound to the actual value function given by α^*_δ. This follows because $\alpha^{*(y)}_{n+1,\delta}$ is chosen optimally for only a subset of $\mathcal{Y}$ (at the points $\theta(y)$), and for all other $y \in \mathcal{Y}$, $\alpha_{n+1,\delta}$ is suboptimal, producing a lower value.

Next, we can bound $\alpha^{*(\theta(y))}_{n+1,\delta} \tau_y(dy|z', u)$ from below based on how the points w are defined.

$$
\begin{aligned}
|V^*_{n,\delta}(\sigma_\delta) - \hat{V}^*_{n,\delta}(\sigma_\delta)| &\leq \int_{\overline{K}} \left[\alpha^{*(y)}_{n+1,\delta}(z')\tau_y(y|z',u^*) - \alpha^{*(\theta(y))}_{n+1,\delta}(z')\tau_y(\theta(y)|z',u)\right] dy \\
&\quad + \frac{\epsilon}{N} \\
&\leq N_q\overline{\lambda}h^{(1)}_y\delta^y + \frac{\epsilon}{N}
\end{aligned}
$$

■

Lemma 5.10 requires defining the representative points $w = (y^{x,i,y^q}, y^q)$ so that the α-vectors satisfy an inequality. Without this requirement, finding $\alpha^{*(w)}_{n+1,\delta}$ at a finite set of points still guarantees a lower bound to the value function for any time n, and is intuitively more accurate as $\delta^y \rightarrow 0$.

Lemma 5.10 leads to the following theorem regarding the error between $V^*_{n,\delta}(\sigma_\delta)$ (based on continuous observations) and $\tilde{V}_{n,\delta}(\sigma_\delta)$ (based on discretized observations). We again use the notation $\tilde{V}$ to indicate that $\tilde{V}$ is represented by the set $\tilde{\Gamma}$ of α-functions calculated using the discretized observations.

Theorem 5.6. *Given discretized observation process W_δ with transition function (5.83), for any time $n \in [0, N]$, the error between $V^*_{n,\delta}(\sigma_\delta)$ calculated according to $\mathcal{Y}$ and $\tilde{V}^*_{n,\delta}(\sigma_\delta)$ calculated according to W_δ satisfies*

$$
0 \leq V^*_{n,\delta}(\sigma_\delta) - \tilde{V}^*_{n,\delta}(\sigma_\delta) \leq (N-n)N_q\overline{\lambda}h^{(1)}_y\delta^y + \frac{(N-n)\epsilon}{N}
$$

for any $\sigma_\delta \in \Sigma_\delta$, with $\overline{\lambda}$ the largest Lebesgue measure of sets $\overline{K}_{y^q}$.

Specifically, the viability probability for $\hat{\mathcal{H}}^{po}$ satisfies

$$
p^N_{\text{viab}}(\rho_\delta; K_\delta) - \tilde{V}^*_{0,\delta}(\rho_\delta) \leq NN_q\overline{\lambda}h^{(1)}_y\delta^y + \epsilon.
$$

Proof. By induction. At time N, $V^*_{N,\delta}(\sigma_\delta) = \tilde{V}^*_{N,\delta}(\sigma_\delta)$ since $\Gamma_N = \tilde{\Gamma}_N = 1_{K_\delta}$. Assume for all $i = n+1, \ldots, N-1$ that $V^*_{i,\delta}(\sigma_\delta) - \tilde{V}^*_{i,\delta}(\sigma_\delta) \leq (N-i)N_q\overline{\lambda}h^{(1)}_y\delta^y + \frac{(N-j)\epsilon}{N}$. Then, at time n,

$$\begin{aligned}
|V^*_{n,\delta}(\sigma_\delta) - \tilde{V}^*_{n,\delta}(\sigma_\delta)| &= \langle \alpha^*_{n,\delta}, \sigma_\delta\rangle - \langle \tilde{\alpha}^*_{n,\delta}, \sigma_\delta\rangle \\
&= \langle \alpha^*_{n,\delta}, \sigma_\delta\rangle - \langle \hat{\alpha}^*_{n,\delta}, \sigma_\delta\rangle + \langle \hat{\alpha}^*_{n,\delta}, \sigma_\delta\rangle - \langle \tilde{\alpha}^*_{n,\delta}, \sigma_\delta\rangle \\
&\leq N_q \overline{\lambda} h^{(1)}_y \delta^y + \int_{\mathcal{Y}} \sup_{\Gamma_{n+1,\delta}} \langle \alpha_{n+1,\delta}, \Phi^\delta_{y,\hat{u}^*} \tilde{\sigma}_\delta\rangle \mathbb{P}(dy|\sigma_\delta, \hat{u}^*) \\
&\qquad - \int_{\mathcal{Y}} \sup_{\tilde{\Gamma}_{n+1,\delta}} \langle \tilde{\alpha}_{n+1,\delta}, \Phi^\delta_{y,\tilde{u}^*} \tilde{\sigma}_\delta\rangle \mathbb{P}(dy|\sigma_\delta, \tilde{u}^*) + \frac{\epsilon}{N} \\
&\leq N_q \overline{\lambda} h^{(1)}_y \delta^y + V^*_{n+1,\delta}(\sigma_{n+1,\delta}) - \tilde{V}^*_{n+1,\delta}(\sigma_{n+1,\delta}) + \frac{\epsilon}{N} \\
&\leq N_q \overline{\lambda} h^{(1)}_y \delta^y + \frac{\epsilon}{N} \\
&\qquad + (N-n-1) N_q \overline{\lambda} h^{(1)}_y \delta^y + \frac{(N-n-1)\epsilon}{N}
\end{aligned}$$

Combining terms completes the proof. ■

Algorithm 5.1 summarizes the procedure for numerically solving Problem 5.1 using the finite state approximation $\hat{\mathcal{H}}^{po}$, discretized observations, and a sampled set $\tilde{\Sigma} \subset \Sigma$. The algorithm assumes the sets Z_δ, W_δ, and $\tilde{\Sigma}$ have already been generated.

Discretized observations for Gaussian mixture approximation

The results of discretizing the observations for the Gaussian mixture abstraction are nearly identical to those for the finite state abstraction. The main difference arises in approximating the integral $\tau_y(\Theta(w)|s', u)$ with a Gaussian sum: To ensure the approximate value function provides a lower bound to $V^*_{n,g}$, we must under-approximate the integral $\tau_y(\Theta(w))$ for each w. We again define $\tilde{V}^*_{n,g}$ similarly to $\tilde{V}^*_{n,\delta}$, with

$$\tilde{\alpha}_{n,g}(s) = \sum_{w \in W_\delta} \tilde{\alpha}^g_{w,u,\sigma}(s) \tag{5.89}$$

$$\tilde{\alpha}^g_{w,u,\sigma}(s) = \int_{\mathcal{S}} \tilde{\alpha}^{*(w)}_{n+1,g}(s') \tau^g_y(w|s', u) \tau_s(ds'|s, u) \sum_{i=1}^{I_q} w^I_i(q) \phi^I_i(x) \tag{5.90}$$

$$\tilde{\alpha}^{*(w)}_{n+1,g}(s') = \arg\left\{ \sup_{\tilde{\alpha}^i_{n+1,g}\in\tilde{\Gamma}_{n+1,g}} \int_{\mathcal{S}} \tilde{\alpha}^i_{n+1,g}(s')\tau^g_y(w|s',u)\tau_s(ds'|s,u) \right.$$
$$\left. \times \sum_{i=1}^{I_q} w^I_i(q)\phi^I_i(x) \right\} \quad (5.91)$$

and $\hat{V}^*_{n,g}$ is the intermediate value function that finds the optimal $\alpha^{*(w)}_{n+1,g} \in \Gamma_{n+1,g}$, rather than in $\tilde{\Gamma}_{n+1,g}$. We can bound the error between $V^*_{n,g}$ and $\hat{V}^*_{n,g}$, and between $V^*_{n,g}$ and $\tilde{V}^*_{n,g}$, equivalently to Lemma 5.10 and Theorem 5.6, respectively.

Lemma 5.11. *For any time $n \in [0, N]$ and $\sigma_g \in \Sigma_g$, the error between $V^*_{n,g}(\sigma_g)$ and $\hat{V}^*_{n,g}(\sigma_g)$ satisfies*

$$0 \leq V^*_{n,g}(\sigma_g) - \hat{V}^*_{n,g}(\sigma_g) \leq N_q\overline{\lambda}h^{(1)}_y\delta^y + \frac{\epsilon}{N}$$

given that the observations w are chosen so that

$$\int_{\mathcal{S}} \alpha^{*(w)}_{n+1,g}(s')T^g_y(w|s',u)\tau_s(s'|s,u)\sum_{i=1}^{I_q} w^I_i(q')\phi^I_i(x')$$
$$> \int_{\mathcal{S}} \alpha^{*(w)}_{n+1,g}(s')\tau_y(w|s',u)|\Theta(w)|\tau_s(s'|s,u)\sum_{i=1}^{I_q} w^I_i(q')\phi^I_i(x'),$$

and with $\overline{\lambda}$ the largest Lebesgue measure of sets $\overline{K}_{y^q}$.

Theorem 5.7. *Given discretized observation process W_δ with transition function (5.84), for any time $n \in [0, N]$, the error between $V^*_{n,g}(\sigma_g)$ calculated according to $\mathcal{Y}$ and $\tilde{V}^*_{n,g}(\sigma_g)$ calculated according to W_δ satisfies*

$$0 \leq V^*_{n,g}(\sigma_g) - \tilde{V}^*_{n,g}(\sigma_g) \leq (N-n)N_q\overline{\lambda}h^{(1)}_y\delta^y + \frac{(N-n)\epsilon}{N}$$

for any $\sigma_g \in \Sigma_g$, with $\overline{\lambda}$ the largest Lebesgue measure of sets $\overline{K}_{y^q}$.

Specifically, the viability probability for the Gaussian mixture approximation satisfies

$$p^N_{\text{viab}}(\rho_g; K_g) - \tilde{V}^*_{0,g}(\rho_g) \leq NN_q\overline{\lambda}h^{(1)}_y\delta^y + \epsilon.$$

The proofs of Lemma 5.11 and Theorem 5.7 follow directly from the proofs of Lemma 5.10 and Theorem 5.6.

Algorithm 5.1 PBVI for Finite State Approximation to Viability

Input: $\mathcal{H}^{po}$, $\hat{\mathcal{H}}^{po}$, W_δ, τ_y^δ, $\tilde{\Sigma}$, safe set K, terminal time N

Output: Estimate $\tilde{V}_{0,\delta}^*(\rho_\delta)$ to viability probability $V_0^*(\rho)$

```
n = N
α = 1_K, Γ̃_n = {α}
for n = N − 1; n ≥ 0; n = n − 1 do
    Γ̃_n = ∅
    for all σ ∈ Σ̃ do
        for u ∈ 𝒰 do
            for w ∈ 𝒴̃_δ do
                Calculate $\tilde{\alpha}_{w,u,\sigma}^\delta$ according to (5.86)
            end for
            $\tilde{\alpha}_{u,\sigma}^\delta = \sum_{w \in \tilde{\mathcal{Y}}_\delta} \tilde{\alpha}_{w,u,\sigma}^\delta$
        end for
        $\tilde{\alpha}_\sigma^\delta = \arg\max_{\tilde{\alpha}_{u,\sigma}^\delta} \langle \tilde{\alpha}_{u,\sigma}^\delta, \sigma \rangle$
        if $\tilde{\alpha}_\sigma^\delta \notin \tilde{\Gamma}_n$ then
            $\tilde{\Gamma}_n = \tilde{\Gamma}_n \cup \{\tilde{\alpha}_\sigma^\delta\}$
        end if
    end for
end for
```

Algorithm 5.2 summarizes the procedure for numerically solving Problem 5.1 using the Gaussian mixture approximation, discretized observations, and a sampled set $\tilde{\Sigma} \subset \Sigma$. The algorithm assumes the RBF approximation $\{w_i(q), \phi_i(x)\}_{i=1}^{I_q}$, W_δ, and $\tilde{\Sigma}$ have already been generated.

To summarize, given either the finite state or Gaussian mixture approximation, we can subsequently 1) sample y from $\mathcal{Y}$ and u from $\mathcal{U}$ to generate a subset $\tilde{\Sigma}_\delta$ or $\tilde{\Sigma}_g$, and 2) discretize $\mathcal{Y}$ and use the set W_{δ^y} to calculate $\tilde{\alpha}_{w,u,\sigma_\delta}^\delta$ or $\tilde{\alpha}_{w,u,\sigma_g}^g$, which are then used to generate $\tilde{\alpha}_{n,\delta} \in \tilde{\Gamma}_{n,\delta}$ and $\tilde{\alpha}_{n,g} \in \tilde{\Gamma}_{n,g}$. Using sets $\tilde{\Sigma}_{n,\delta}$ and $\tilde{\Gamma}_{n,\delta}$ in place

Algorithm 5.2 PBVI for Gaussian Mixture Approximation to Viability

Input: $\mathscr{H}^{po}$, $\{w_i(q), \phi_i(x)\}_{i=1}^{I_q}$, W_δ, τ_y^g, $\tilde{\Sigma}$, safe set K, terminal time N

Output: Estimate $\tilde{V}_{0,g}^*(\rho_g)$ to safety probability $V_0^*(\rho)$

```
$n = N$
$\alpha = \{w_i(q), \phi_i(x)\}_{i=1}^{I_q}$, $\tilde{\Gamma}_n = \{\alpha\}$
for $n = N-1$; $n \geq 0$; $n = n-1$ do
    $\tilde{\Gamma}_n = \emptyset$
    for all $\sigma \in \tilde{\Sigma}$ do
        for $u \in \mathcal{U}$ do
            for $w \in \tilde{\mathcal{Y}}_\delta$ do
                Calculate $\tilde{\alpha}_{w,u,\sigma}^g$ according to (5.90)
            end for
            $\tilde{\alpha}_{u,\sigma}^g = \sum_{w \in \tilde{\mathcal{Y}}_\delta} \tilde{\alpha}_{w,u,\sigma}^g$
        end for
        $\tilde{\alpha}_\sigma^g = \arg\max_{\tilde{\alpha}_{u,\sigma}^g} \langle \tilde{\alpha}_{u,\sigma}^g, \sigma \rangle$
        if $\alpha_\sigma \notin \Gamma_n$ then
            $\tilde{\Gamma}_n = \tilde{\Gamma}_n \cup \{\alpha_\sigma\}$
        end if
    end for
end for
```

of $\Sigma_{n,\delta}$ and $\Gamma_{n,\delta}$ provides a lower bound to the viability probability $p_{\text{viab}}^N(\rho_\delta; K_\delta)$ that converges to $p_{\text{viab}}^N(\rho_\delta; K_\delta)$ as δ^σ and δ^y approach zero (and similarly for $\tilde{\Sigma}_{n,g}$ and $\tilde{\Gamma}_{n,g}$).

5.7 Extension to Reachability and Reach-Avoid

We can use the alternate information state and value function presented in Chapter 4 for the reach-avoid problem to apply an approximation method similar to that for the viability problem presented above. Adjustments must first be made in the

α-function representation, dependent on the different value function. A similar finite state approximation can be utilized for the reach-avoid problem, where rather than discretizing over K, we discretize over $K \cup T$. For the Gaussian mixture approximation, we will require two separate RBF approximations, one for 1_K, and one for 1_T. Once the above adjustments are made, the same type of PBVI algorithm (Algorithm 5.1 or 5.2) can be used.

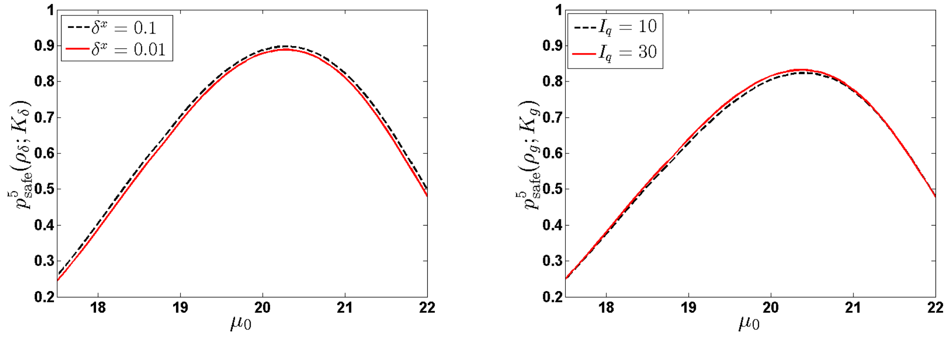
The reachability problem is more difficult if formulated as a type of reach-avoid problem, in that all of $\mathcal{S}$ must be discretized and a finite representation is not possible. However, by representing the reachability problem as the dual of the viability problem, we can solve an equivalent viability problem to generate estimates of reachability probabilities. Recall the relation

$$p^N_{\text{reach}}(\rho; T) = 1 - \inf_{\pi \in \Pi} p^N_{\text{viab}}(\pi, \rho; T^c) \tag{5.92}$$

given in Chapter 2. Then we can apply the approximation methods for the viability problem to generate reachability probabilities, by changing all maximizations to minimizations when calculating the α-functions. However, we require T^c, T, and K to all be compact sets. Otherwise the discretization procedure for both the state space $\mathcal{S}$ and observation space $\mathcal{Y}$ will have to be altered.

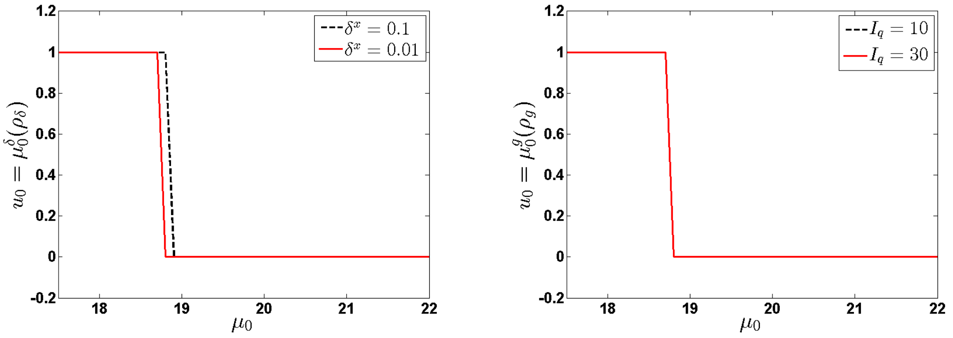
5.8 Numerical Example

We again consider the temperature regulation problem presented in Chapter 5. We consider the case of one heater, which can either be turned on to heat one room, or turned off. The temperature of the room at time n is given by the continuous variable x_n, and the discrete state $q_n = 1$ indicates the heater is on at time n, and $q_n = 0$ denotes the heater is off. The stochastic difference equation governing the temperature is given in (4.40) with the number of rooms $M = 1$. The constants b_1, c_1, and x_a are set to $b_1 = 0.0167$, $c_1 = 0.8$, and $x_a = 6$, and $v_n \in \mathbb{R}$ are i.i.d. Gaussian random variables with mean zero and variance v^2. The control input is given by $u_n \in \mathcal{U}$ with $\mathcal{U} = \{0, 1\}$, but the chosen control is not always implemented with probability 1.



(a) (b)

Figure 5.1: Comparison of viability probabilities over varying initial distribution $\rho = \phi(x; \mu_0, 1)$ and $q_0 = 0$, using the finite state approximation (a) and Gaussian mixture approximation (b). In both (a) and (b) $\delta^y = 0.5$. Fig. (a) compares probabilities for $\delta^x = 0.1$ (black dashed line) and $\delta^x = 0.01$ (red solid line). Fig. (b) compares probabilities for $I_q = 10$ (black dashed line) and $I_q = 30$ (red solid line). The refinement of δ^x and increase in I_q has a small impact on the viability probabilities. The finite state approximation estimates higher probabilities for μ_0 in the interior of K than the Gaussian mixture approximation.



(a) (b)

Figure 5.2: Comparison of optimal control inputs as a function of $\rho = \phi(x; \mu_0, 1)$ with $q_0 = 0$, using the finite state approximation (a) and Gaussian mixture approximation (b). In both (a) and (b), $\delta^y = 0.5$. Fig. (a) compares control inputs for $\delta^x = 0.1$ (black dashed line) and $\delta^x = 0.01$ (red solid line). Fig. (b) compares control inputs for $I_q = 10$ (black dashed line) and $I_q = 30$ (red solid line), which in this case are the same. All approaches produce a thresh-hold policy that turns the heater off for $\mu_0 > 18.7$, except the finite approximation with $\delta^x = 0.1$, which turns the heater off for $\mu_0 > 18.8$.

Instead, q_n is updated probabilistically, dependent on u_{n-1} and q_{n-1}, with transition function $T_q(q_{n+1}|q_n, u_n)$. So while function $\mu_n(\sigma_n)$ deterministically returns a single control input, control input $u_n = \mu_n(\sigma_n)$ may not always be implemented.

As in the previous chapter, we assume the actual temperature is unknown, and only a noisy measurement is available to the controller. The controller does, however, know whether the heater is on or off at time n (i.e. q_n is perfectly observed). The observation $y_n = y_n^x$ is given by $y_n^x = x_n + w_n$, with w_n i.i.d. Gaussian random variables with mean zero and variance w^2.

It is desirable to keep the temperature of the room between 17.5 and 22 degrees Celsius at all times, hence the safe region $K = [17.5, 22]$ does not depend on the discrete state q_n (so $1_K(s) = 1_K(x)$). We consider the viability probability of remaining within K for $N = 5$ time steps given initial temperature distribution ρ normally distributed with varying mean $\mu_0 \in K$ and variance $\Sigma_0 = 1$. The initial mode is given as $q_0 = 0$. The finite state and Gaussian mixture approximations are used in a PBVI algorithm in the style of Perseus [PVSP06].

We consider a uniform grid ($\delta_{i,q}^x = \delta^x$ constant for all i, q) over the region $K \subset \mathbb{R}$ for the finite state approximation, with representative points at the end-point of each grid cell. For example, setting $\delta^x = 0.1$ gives $x^{1,q} = 17.5, x^{2,q} = 17.6, \dots$ for $q = 0$ and $q = 1$, and a total of $m_q = 45$ cells $K_{i,q}$. The function $\xi(x, q)$ maps q to itself, and maps x to the nearest $x^{i,q}$ in absolute value.

The Gaussian mixture approximation utilizes an RBF approximation of the indicator function calculated using MATLAB's *gmdistribution* function. We used a reduction process to limit the number of components of each α and σ for the Gaussian mixture approximation. Similar Gaussians are combined into a single component based on the L^2 distance between functions [ZK10]. Each mixture was limited to 30 components to reduce overall computation time without overly sacrificing accuracy. This number can easily be changed, however, depending on the importance of speed versus accuracy.

Both approximations employ a sampled set $\tilde{\Sigma}$ and a finite set of observations to calculate the α-functions for the PBVI algorithm. To generate the set $\tilde{\Sigma}$, we initialized a set of 40 states σ_0 normally distributed with variance Σ_0 and mean μ_0 randomly chosen uniformly on K. Each σ_0 was updated according to $\Phi^g_{y,u}$ with u chosen randomly and y sampled from the corresponding σ_0 (i.e. $y \sim \mathbb{P}(y|\sigma_0, u)$). This process was repeated N times, so that for each time step we had a set of 40 sampled σs. The finite set of observations were produced by a uniform grid over $\overline{K} = [16, 24]$, again using end-points as the representative observations.

To compare performance of the finite state and Gaussian mixture approximations, we present computation times and viability probability estimates for each, with varying δ^x, δ^y, and number of components in the indicator approximation. Viability probabilities for varying initial distributions ρ are presented in Figs. 5.1a and 5.1b for the finite state approximation and Gaussian mixture approximation, respectively. The optimal policy at time zero is shown for varying ρ in Figs. 5.2a and 5.2b for the finite and Gaussian approximations, respectively. Computation times for the finite state approximation are given in Table 5.1, and for the Gaussian mixture approximation in Table 5.2.

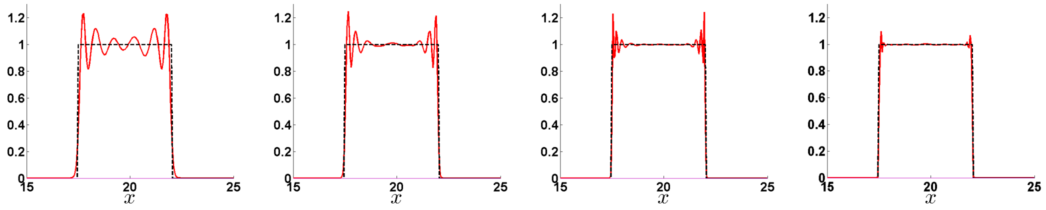
	$\delta^x = 0.1$			$\delta^x = 0.01$		
	$\delta^y = 1.0$	$\delta^y = 0.5$	$\delta^y = 0.1$	$\delta^y = 1.0$	$\delta^y = 0.5$	$\delta^y = 0.1$
Comp. time (s)	50.5	205.1	1599.8	8961.1	15343.7	108591.3

Table 5.1: Computation times using PBVI with finite state approximation, for varying continuous state spacing δ^x and discretized observation spacing δ^y.

We also show sample Gaussian mixture approximations to the indicator function 1_K in Fig. 5.3 with varying numbers of components I_q. The L^1 error between the RBF approximation and 1_K for varying I_q is shown in Fig. 5.4. As the number of components increases, the approximation becomes more accurate, although as seen in Fig. 5.3, oscillations remain at the endpoints of K. The increasing accuracy is most apparent in Fig. 5.4, and demonstrates the convergence towards zero of the L^1 error with increasing I_q.

	$I_q = 10$		$I_q = 30$	
	$\delta^y = 1.0$	$\delta^y = 0.5$	$\delta^y = 1.0$	$\delta^y = 0.5$
Comp. time (s)	365.5	1625.9	1865.0	5586.1

Table 5.2: Computation times using PBVI with Gaussian mixture approximation, for varying number of components I_q for RBF approximation to 1_K and discretized observation spacing δ^y.



(a) (b) (c) (d)

Figure 5.3: Comparison between $1_K(x)$ (in black, dashed line) to RBF approximation (red, solid line) for (a) $I_q = 10$ components, (b) $I_q = 30$ components, (c) $I_q = 100$ components, and (d) $I_q = 400$ components. As the number of components increases, the approximation improves, although oscillations at the endpoints remain.

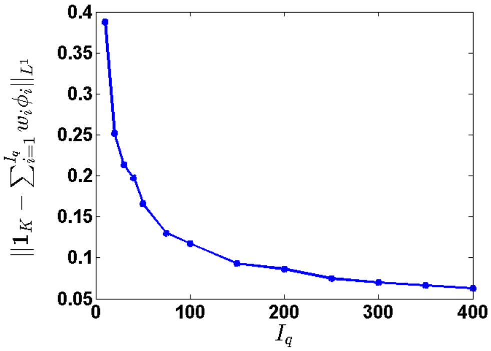


Figure 5.4: The L^1 error for RBF approximations to indicator function 1_K with a varying number of components I_q. As the number of components increases, the error converges towards zero.

We show viability probabilities for $\delta^y = 0.5$ only, because the change in viability estimates when varying δ^y is not noticeable enough to merit comparison. Decreasing δ^y causes a slight increase in the viability probabilities, as expected, but there is not a significant improvement in the probability estimates, although as seen in Tables 5.1

and 5.2, the increase in computation time is significant. This is likely problem-specific, and the value of δ^y may have a greater impact for some applications.

The viability probability estimates for the finite state approximation are in general greater than for the Gaussian mixture approximation. The mixture reduction method employed, as well as the indicator function approximation, make the Gaussian method seemingly less accurate than the finite state approximation. However, over a finer mesh δ^x, the finite state method results in greater computation time. Although the coarse grid produces similar results to the fine grid ($\delta^x = 0.1$ versus $\delta^x = 0.01$), in higher dimensional problems the number of grid cells becomes prohibitive even when δ^x is large. All scenarios produce a nearly identical optimal thresh-hold policy based on the initial mean μ_0, indicating that an optimal policy may be computed fairly quickly using any of the above methods.

Interestingly, increasing the number of components in the RBF approximation to the indicator function only slightly improves the viability estimates of the Gaussian mixture approximation, although the L^1 error from increasing the number of components to 30 drops significantly. This may be caused by the mixture reduction technique, leading to a loss in the added benefit of an increased number of components when that number is again reduced. However, although the L^1 error with $I_q = 10$ is large, we obtain viability estimates that are quite similar to the finite state approximation. This requires further investigation, but may help in decreasing computation time without losing significant accuracy by choosing I_q to be small.

5.9 Summary

We have presented the first numerical results for verification of a partially observable DTSHS, via two approximations that enable the use of a well-known POMDP optimization technique. The first approximation discretizes the state space over a compact set K and enables a vector representation of the information states and α-functions. The second approximates the indicator function over compact set K using a finite set of Gaussian radial basis functions and enables a Gaussian mixture representation of the information states and α-functions. We can apply point-based value iteration to either approximation, and guarantee a lower bound to the viability probability, which is proven to converge to the true viability probability of the original PODTSHS. A simple numerical example shows that both methods provide similar viability estimates. The finite state approximation is faster when a coarse discretization is used, but quickly becomes slower than the Gaussian mixture approximation with a finer discretization. Although we present a linear system with additive Gaussian noise, both approximations may be extended to non-Gaussian systems. Convergence results for the finite state approximation apply to arbitrary transition kernels T_x and T_y. The Gaussian mixture approximation further requires approximating T_x and T_y with Gaussian mixtures, and introduces additional error.

Chapter 6

Linear Time-Invariant Systems with Noisy Observations

This chapter considers linear time-invariant (LTI) systems with noisy state measurements, and develops an improved algorithm based on the PBVI approach of the previous chapter for computing reachability probabilities tailored to the class of LTI systems. For a linear system with Gaussian measurement noise and without process noise, the information state is shown to be a truncated Gaussian, and a novel PBVI algorithm $PointBasedSafety$ is proposed that extends existing point-based solvers to include the truncated Gaussian information state. We also extend the generic discretization procedure for the observation space from the previous chapter to an adaptive grid scheme that reduces estimation error and increases speed of computation. Preliminary results show the method to be promising in terms of computation time as compared to other approaches.

6.1 Introduction

Chapter 5 provided two systematic ways of numerically approximating probabilistic reachable sets for a PODTSHS. While we applied both methods to a simple temperature regulation problem, the lengthy computation times indicate the need for specialized algorithms in order to make either approach feasible on higher-dimensional problems. We therefore consider a special class of system with LTI dynamics, no process noise (i.e. the state of the system evolves deterministically), and with an observation process that is a linear function of the current state with additive Gaussian noise. LTI systems, although seemingly simple, have a wide range of applications; even the spacecraft relative motion dynamics of Chapter 3 are modeled as an LTI system. Further, although we do not consider hybrid dynamics in this chapter, we hope the improved algorithm for LTI dynamics may be extended to piecewise-linear systems, which have even greater modeling power.

The main benefit of considering an LTI system with Gaussian measurement noise and no process noise is that the information state can be modeled as a truncated Gaussian, i.e. a Gaussian density that is nonzero only over a subset of $\mathbb{R}^n$, rather than all of $\mathbb{R}^n$. As such, we do not require storage of a large vector of values, nor do we need to store a large collections of weights, means and covariances as with a Gaussian mixture. Updating the information state to incorporate new observations and control inputs is also made simpler, by only requiring three closed form expressions for the updated mean, covariance, and support of the Gaussian, where the support is the region over which the Gaussian is nonzero.

We also implement an adaptive gridding scheme for the observations, inspired by a similar scheme presented in [SA13]. Whereas in the previous chapter we presented a generic means of dividing a superset of the safe region K into cells, each of which has a representative point (y^x, y^q), we now generate a grid of cells that varies in number, and with nonuniform cell sizes. The size and shape of the cells are determined repeatedly in the new point-based algorithm, once for each backup operation (each time we estimate the value function at a different sampled information state). By varying the size and shape of the cells, we are able to increase the speed of the algorithm while simultaneously decreasing the error of the value function approximation.

To summarize, the contributions of this chapter are threefold. First, while in general the information state for the viability problem does not have a closed form representation, we show that for an LTI system with Gaussian measurement noise, the information state is a truncated Gaussian. Second, instead of discretizing the observation space explicitly, we generate a grid over the observation space with non-uniform cell sizes, for each information state that we sample, and derive an error bound on the value function. Finally, we present a unique point-based algorithm that combines the continuous belief state with a vector representation of the value function, leading to some of the first numerical results for safety calculations in partially observable domains. We again focus on the viability problem, because as shown in previous chapters, extensions to reachability and reach-avoid problems are implicit.

The rest of the chapter is organized as follows. We describe the LTI system model and simplified dynamic programming recursion in Section 6.2. Section 6.3 proves that the information state is a truncated Gaussian, and describes the adaptive gridding procedure, including an error bound on the value function approximation when using the nonuniform grid. It also gives the updated point-based algorithm for an LTI system. In section 6.4 we provide two numerical examples that demonstrate the improved performance of the algorithm. The first is a two-dimensional temperature regulation problem modified from the previous chapter to include two separate rooms to be heated. We use this example to compare the performance of our algorithm with other variations of the point-based algorithm that either uniformly grid the observation space, or take a full discretization approach as in the previous chapter. The second example concerns automated anesthesia delivery, modeled as a three-dimensional system.

6.2 Problem Formulation

6.2.1 System Model

We consider a discrete time dynamical system with state space $\mathcal{X} \subseteq \mathbb{R}^n$, finite control space $\mathcal{U} \subset \mathbb{R}^n$, and linear time-invariant (LTI) dynamics given by

$$x_{n+1} = Ax_n + g(u_n) \tag{6.1}$$

in which we assume A is an invertible $n \times n$ matrix and g is a function mapping $\mathcal{U}$ to $\mathcal{R}^n$. We use x rather than s to represent the state of the system, to emphasize that we are not considering a hybrid state space. Further, the true state x is unknown to the controller, and only a noisy observation is available,

$$y_n = Cx_n + w_n \tag{6.2}$$

with $y_n \in \mathcal{Y} = \mathbb{R}^n$, and C an invertible $n \times n$ matrix. The variable w_n is assumed Gausian with zero mean and diagonal covariance matrix $\mathcal{W} = diag\{w^{(i)}\}$, $w_n \sim \mathcal{N}(0, \mathcal{W})$. Finally, the initial state x_0 is assumed unknown but Gaussian, $x_0 \sim \rho(x_0) = \mathcal{N}(\mu_0, \Sigma_0)$.

Given dynamics (6.1) and measurements (6.2), we wish to determine whether the state x of the system can remain within some predefined safe region $K \subseteq \mathcal{B}(\mathcal{X})$ ($\mathcal{B}(\mathcal{X})$ denoting the Borel σ-algebra on $\mathcal{X}$), over a finite time horizon N.

Recall that the viability probability is given by

$$p^N_{\text{viab}}(\rho; K) = \sup_{\pi \in \Pi} \mathbb{E}^\pi \left[\prod_{n=0}^{N} 1_K(x_n) \middle| \rho \right]. \tag{6.3}$$

The problem we wish to solve, modified from Chapter 5 to incorporate the simplified dynamics, is as follows.

Problem 6.1. *Consider a system that evolves according to* (6.1)*, with measurements* (6.2) *and initial Gaussian distribution $\rho(x)$. Given a safe set K and time horizon N:*

1. *Compute the maximal probability* (6.3) *of remaining within K for N time steps.*

2. Compute the optimal policy π^* that maximizes (6.3).

If the maximal probability and optimal policy cannot be computed exactly (which is quite likely [LGM01]), an approximation that produces a suboptimal policy and lower bound on the maximal safety probability is desired.

6.2.2 Dynamic Programming Solution

We again present an information state encapsulating all available information for making control decisions, as well as a dynamic programming recursion that uses a value function evaluated over the information state. We modify the operator Φ for updating the information state, and the value function for the dynamic program, to reflect the LTI dynamics (6.1) and observation process (6.2).

We use $\phi(x; \mu, \mathcal{S})$ to denote a normal probability density function evaluated at x with mean μ and covariance $\mathcal{S}$. The function τ_s, a stochastic transition kernel for the state variable x, is now an impulse function

$$\tau_s(x'|x, u) = \delta(x' - Ax - g(u)). \tag{6.4}$$

The information state is defined as

$$\begin{aligned} \sigma_0 &= \rho \\ \sigma_n &= \Phi_{y_n, u_{n-1}} \sigma_{n-1} \end{aligned} \tag{6.5}$$

with $\Phi_{y,u}\sigma$ given by

$$\Phi_{y,u}\sigma(x') = \beta\phi(y; Cx', \mathcal{W}) \int_{\mathcal{X}} 1_K(x)\delta(x' - Ax - g(u))\sigma(x)\, dx \tag{6.6}$$

and β is a normalizing constant. The information state σ therefore lies in the space of probability distributions on $\mathcal{X}$, $\sigma : \mathcal{X} \to [0, 1]$.

The dynamic programming recursion over the information state σ for Problem 6.1

is

$$\begin{aligned} V_N^*(\sigma) &= \langle \sigma, 1_K \rangle \\ V_n^*(\sigma) &= \max_{u\in\mathcal{U}} \int_{\mathcal{Y}} V_{n+1}^*(\Phi_{y,u}\sigma)\mathbb{P}(dy|\sigma,u) \\ &= \max_{u\in\mathcal{U}} \int_{\mathcal{Y}} \int_K V_{n+1}^*(\Phi_{y,u}\sigma)\phi(dy; C(Ax+g(u)),\mathcal{W}))\sigma(dx) \end{aligned} \tag{6.7}$$

resulting in $V_0^*(\rho) = p_{viab}^N(\rho; K)$. The value function still admits an α-function representation, such that for all n, $V_n^*(\sigma) = \sup_{\alpha_n^i \in \Gamma_n} \langle \alpha_n^i, \sigma \rangle$ with

$$\alpha_n^i(x) = \int_{\mathcal{Y}} \alpha_{y,u,\sigma^i}(x)\, dy \tag{6.8}$$

$$\alpha_{y,u,\sigma}(x) = \alpha_{n+1}^{*(y)}(Ax+g(u))\phi(y; C(Ax+g(u)),\mathcal{W})1_K(x) \tag{6.9}$$

$$\alpha_{n+1}^{*(y)}(x') = \arg\left\{ \sup_{\alpha_{n+1}^i \in \Gamma_{n+1}} \int_K \alpha_{n+1}^i(Ax+g(u))\phi(y; C(Ax+g(u)),\mathcal{W})\sigma(dx) \right\} \tag{6.10}$$

We now consider approximately solving Problem 6.1 again using point-based value iteration (PBVI).

6.3 PBVI for Verification of an LTI System

6.3.1 Information State

Ideally, we would like to represent the information state σ as a Gaussian, since a Gaussian can be characterized solely through its mean and covariance, and Gaussians are preserved under multiplication and integration. The presence of the indicator function in (6.6) unfortunately ruins any hope of maintaining σ as a Gaussian, even when ρ, w_n and τ are assumed Gaussian.

However, in the case in which τ is a delta function (i.e. deterministic state dynamics), the information state can be represented by a *truncated* Gaussian, meaning a Gaussian that is nonzero only over a subset of $\mathbb{R}^n$. We will denote a truncated

Gaussian by $\phi(x;\mu,\Sigma,I)$ where I is the interval over which ϕ is nonzero (also known as the *support* of ϕ). The nature of the sufficient statistic does not, however, guarantee that the integral of the truncated Gaussian over support I is equal to one. Instead, $\sigma_n(x)$ is a scaled, truncated Gaussian that integrates to one over a superset of I, as we will demonstrate.

Proposition 6.1. *Given dynamics (6.1) and measurements (6.2), τ defined as in (6.4), $\rho(x_0)$ Gaussian, and update equation (6.6), σ_n is given by*

$$\sigma_n(x) = \beta_n \phi(x;\mu_n,\Sigma_n,I_n)$$

for all $n = 0,\ldots,N$. The parameters μ_n, Σ_n, and I_n are given by

$$\begin{aligned} \mu_n &= \Sigma_n \left(\left(C^{-1}\mathcal{W}C^{-T} \right)^{-1} Cy + \left(A\Sigma_{n-1}A^T \right)^{-1} \left(A\mu_{n-1} + g(u) \right) \right), \\ \Sigma_n &= \left(\left(C^{-1}\mathcal{W}C^{-T} \right)^{-1} + \left(A\Sigma_{n-1}A^T \right)^{-1} \right)^{-1}, \\ I_n &= \left\{ x : A^{-1}(x - g(u)) \in K \cap I_{n-1} \right\}. \end{aligned}$$

The scaling factor $\beta_n = \frac{1}{\mathbb{P}(y|\sigma_{n-1},u)}$ ensures that $\int_{\tilde{I}_n} \sigma_n(dx) = 1$, with $\tilde{I}_n = \{x : A^{-1}(x - g(u)) \in I_{n-1}\}$.

Proof. By induction. At time 0, $\sigma_0 = \rho$ is normally distributed by definition, with mean μ_0, covariance Σ_0, and support $I_0 = \mathbb{R}^n$. At time $n-1$, we assume $\sigma_{n-1}(x) = \beta_{n-1}\phi(x;\mu_{n-1},\Sigma_{n-1},I_{n-1})$. Then according to (6.5), for observation y and control input u, we obtain

$$\sigma_n(x') = \frac{\beta_{n-1}}{\mathbb{P}(y|\sigma_{n-1},u)} \phi(y;Cx',\mathcal{W}) \int_{\mathcal{X}} 1_K(x)\delta(x' - Ax - g(u))\phi(dx;\mu_{n-1},\Sigma_{n-1},I_{n-1}). \tag{6.11}$$

Expanding gives

$$\sigma_n(x') = \frac{\beta_{n-1}\phi(y;Cx',\mathcal{W}) \int_{\mathcal{X}} 1_K(x)\delta(x' - Ax - g(u))\phi(dx;\mu_{n-1},\Sigma_{n-1},I_{n-1})}{\beta_{n-1} \int_{\mathcal{X}} \phi(y;Cx',\mathcal{W}) \int_{\mathcal{X}} \delta(x' - Ax - g(u))\phi(dx;\mu_{n-1},\Sigma_{n-1},I_{n-1})dx'} \tag{6.12}$$

We first examine the numerator alone (after canceling the terms β_{n-1}), which can be rewritten as

$$\phi(y;Cx',\mathcal{W})\phi(A^{-1}(x' - g(u));\mu_{n-1},\Sigma_{n-1}) \\ \times 1_K(A^{-1}(x' - g(u)))1_{I_{n-1}}(A^{-1}(x' - g(u))). \tag{6.13}$$

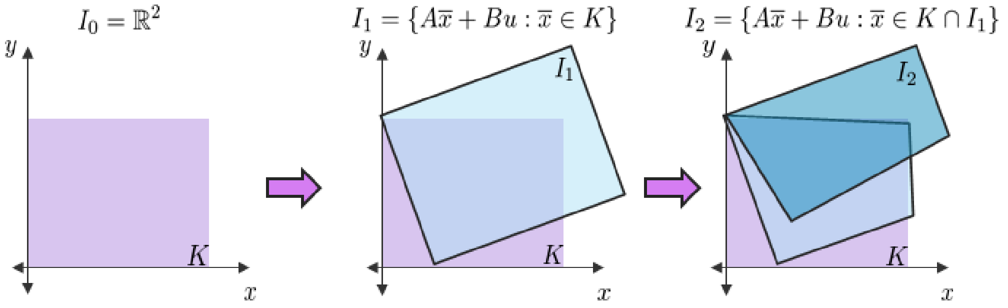

$I_0 = \mathbb{R}^2$
y
K
x
$I_1 = \{A\overline{x} + Bu : \overline{x} \in K\}$
y
I_1
K
x
$I_2 = \{A\overline{x} + Bu : \overline{x} \in K \cap I_1\}$
y
I_2
K
x

Figure 6.1: Example of how truncated region I_n for the information state is updated at each time step, for an LTI system in $\mathbb{R}^2$. At time step n, I_n is intersected with K, and the resulting set is propagated through the dynamics, resulting in $I_{n+1} = \{A\overline{x} + Bu : \overline{x} \in K \cap I_n\}$.

The denominator is written as

$$\int_{\mathcal{X}} \phi(y; Cx', \mathcal{W})\phi(A^{-1}(x' - g(u)); \mu_{n-1}, \Sigma_{n-1})1_{I_{n-1}}(A^{-1}(x' - g(u)))dx'. \tag{6.14}$$

Using the identity $\phi(A^{-1}(x' - g(u)); \mu_{n-1}, \Sigma_{n-1}) = |A^{-1}|\phi(x'; A\mu_{n-1} + g(u), A\Sigma_{n-1}A^T)$, the second Gaussian in both (6.13) and (6.14) is rewritten as a function of x' alone, and the two Gaussians are combined according to known identities to produce another Gaussian up to scaling factor $\kappa = \phi(C^{-1}y; A\mu_{n-1} + g(u), C^{-1}\mathcal{W}C^{-T} + A\mathcal{S}_{n-1}A^T)$.

$$\begin{aligned} \sigma_n(x') &= \frac{\kappa\, \phi(x'; \mu_n, \Sigma_n)1_K(A^{-1}(x' - g(u)))1_{I_{n-1}}(A^{-1}(x' - g(u)))}{\kappa \int_{\mathcal{X}} \phi(x'; \mu_n, \Sigma_n)1_{I_{n-1}}(A^{-1}(x' - g(u)))dx'} && (6.15) \\ &= \beta_n \phi(x'; \mu_n, \Sigma_n, I_n) && (6.16) \end{aligned}$$

with μ_n, Σ_n, and I_n defined as above, and β_n a normalizing factor that guarantees $\int_{\tilde{I}_n} \sigma_n(dx) = 1$. ■

A graphical representation of the propagation of the truncated region I_n of the information state is given in Fig. 6.1, for a system with state $x \in \mathbb{R}^2$ and dynamics $x_{n+1} = Ax_n + Bu_n$, and safe region K a rectangle adjacent to the origin. At each time step, the truncated region I_n is intersected with set K, and the entire resulting set is propagated through the linear dynamics to create the new truncated region I_{n+1}.

Representing the information state σ as a truncated Gaussian requires much less storage than if the state space were discretized and σ were represented as a vector. It also gives an exact representation, and no information is lost through a coarse discrete approximation.

6.3.2 Observations

PBVI iteration schemes typically use a small, discrete set of observations. Tailoring these methods to solve (6.7) requires discretization of continuous observations, however repeated evaluation of (6.9) will increase computation time exponentially in the dimension of y.

We therefore propose an adaptive gridding scheme for the continuous observation space $\mathcal{Y}$. For a single information state σ, it is reasonable to assume that nearby observations y in some neighborhood will produce the same $*(y)$, i.e. will be associated with the same optimal $\alpha_{n+1}^{*(y)}$ in (6.9). Hence we grid $\mathcal{Y}$ into cells and find an upper bound on the error associated with the value function V_n^* when assuming all y in a single cell are associated with the same α_{n+1}^i.

Because $\mathcal{Y} = \mathbb{R}^n$, we consider an expanded set $\overline{K} \supset K$ defined so that the probability of observing a value y for $x \in K$ that is outside of $\overline{K}$ is approximately zero, i.e. $\phi(\mathcal{Y}\backslash\overline{K}; Cx, \mathcal{W}) < \epsilon$, $\epsilon \ll 1$ when $x \in K$. Gaussian density ϕ evaluated over set $\mathcal{Y}\backslash K$ is $\int_{\mathcal{Y}\backslash K} \phi(dy; Cx, \mathcal{W})$.

The set $\overline{K}$ is divided into disjoint subsets $\overline{K}_i$, $\bigcup_{i=1,\ldots m} \overline{K}_i = \overline{K}$. We also define $\psi_y = \overline{K}^c = \mathcal{Y}\backslash\overline{K}$, such that $\bigcup_{i=1,\ldots,m} \overline{K}_i \times \{\psi_y\} = \mathbb{R}^n$. The partition of $\overline{K}$ is denoted $\mathcal{G} = \{\overline{K}_i, i = 1, \ldots, m\}$. Each cell $\overline{K}_i$ has a representative point y^i, and the diameter of partition $\overline{K}_i$ is $\delta_i = \sup_{y \in \overline{K}_i} \|y - y^i\|$. The maximum diameter is $\delta = \max_i \delta^i$. Finally, the function θ maps observation y to its representative value y^i, and Θ is a set-valued mapping from y^i to $\overline{K}_i$.

For a given grid configuration $\mathcal{G}$, $V_{n,\mathcal{G}}^*$ denotes the value function approximation at time n using grid $\mathcal{G}$ to calculate the α-vectors. We write $V_{n,\mathcal{G}}^*(\sigma) = \sup_{\alpha_{n,\mathcal{G}} \in \Gamma_{n,\mathcal{G}}} \langle \alpha_{n,\mathcal{G}}, \sigma \rangle$

with

$$\alpha_{n,\mathcal{G}}(x) = \sum_{i=1}^{m} \alpha^{\mathcal{G}}_{y^i,u,\sigma}(x) + \alpha^{\mathcal{G}}_{\psi_y,u,\sigma}(x) \tag{6.17}$$

for some $\sigma \in \Sigma$, $u \in \mathcal{U}$, and intermediate function

$$\alpha^{\mathcal{G}}_{y^i,u,\sigma}(x) = \int_{\mathcal{X}} \alpha^{*(y^i)}_{n+1}(x')\phi(\Theta(y^i); Cx', \mathcal{W})\tau(dx'|x,u)1_K(x). \tag{6.18}$$

We pick points y^i so that

$$\alpha^{*(y^i)}_{n+1}(x')\phi(\Theta(y^i); Cx', \mathcal{W}) > \alpha^{*(y^i)}_{n+1}(x')\phi(y^i; Cx', \mathcal{W})|\Theta(y^i)|. \tag{6.19}$$

To quantify the error resulting from using grid $\mathcal{G}$, we first describe a Lipschitz property for Gaussian $\phi(y; Cx; \mathcal{W})$ similar to that in [SA13].

Lemma 6.1. *The Gaussian $\phi(y; Cx, \mathcal{W})$ satisfies*

$$|\phi(y; Cx, \mathcal{W}) - \phi(\overline{y}; Cx, \mathcal{W})| \leq h(x)\|y - \overline{y}\|_2 \tag{6.20}$$

for $y, \overline{y} \in \overline{K}_i$, and h piecewise convex.

$$h(x) = \begin{cases} \overline{h} = \frac{1}{(2\pi)^{\frac{n}{2}}|\mathcal{W}|^{\frac{1}{2}}\sqrt{w^*_\sigma}} e^{-\frac{1}{2}} & \text{if } x \in A_i \\ \frac{1}{(2\pi)^{\frac{n}{2}}|\mathcal{W}|^{\frac{1}{2}} w^*_\sigma}\|f_i(x)\| e^{-\frac{\|f_i(x)\|^2}{2w^*_\sigma}} & \text{if } x \notin A_i \end{cases}. \tag{6.21}$$

We define $f_i(x) = \hat{y}^i - Cx$, $\hat{y}^i = \arg\min_{y \in \overline{K}_i} \|y - Cx\|$, and $A_i = \{x : f_i(x) \leq \sqrt{3w^*_\sigma}\}$.

Proof. The Lipschitz constant $\overline{h}$ is the maximum value of the norm of the derivative of ϕ with respect to y:

$$\left\|\frac{\partial \phi}{\partial y}\right\|_2 \leq \frac{1}{(2\pi)^{\frac{n}{2}}|\mathcal{W}|^{\frac{1}{2}} w^*_\sigma}\|y - Cx\|_2 e^{-\frac{\|y-Cx\|_2^2}{2w^*_\sigma}}. \tag{6.22}$$

Since $\|\mathcal{W}\|_2 = w^*_\sigma$, the largest eigenvalue of $\mathcal{W}$, the maximum of (6.22) occurs at $\|y - Cx\| = \sqrt{w^*_\sigma}$. For $y \in \overline{K}_i$, we obtain the upper bound $\overline{h}$ when $\|y - Cx\| < \sqrt{3w^*_\sigma}$, and a tighter bound than $\overline{h}$ when $\|y - Cx\| \geq \sqrt{3w^*_\sigma}$ (the tighter bound is also a convex function of $\|y - Cx\|$ for $\|y - Cx\| > \sqrt{3w^*_\sigma}$). We can therefore modify the Lipschitz bound to be a piecewise convex function of x, for $y, \overline{y} \in \overline{K}_i$, in terms of the derivative of ϕ. ■

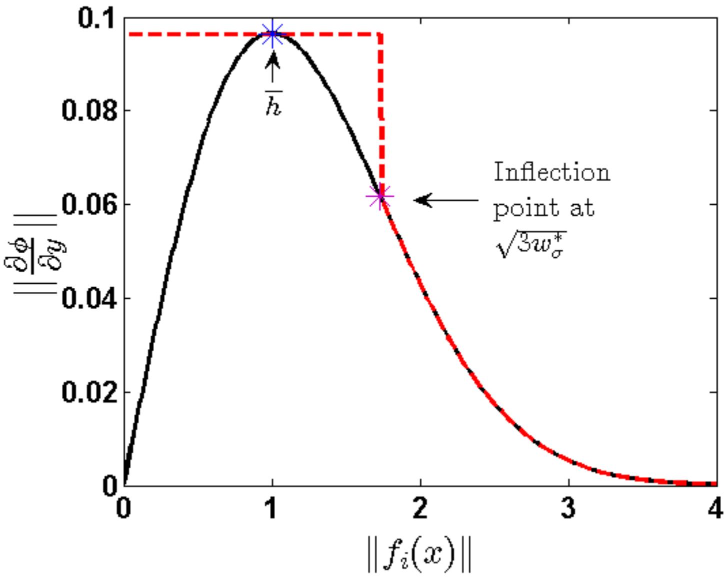


Figure 6.2: The function $\|\frac{\partial\phi}{\partial y}\|$ (in black), expressed for clarity as a function of $\|f_i(x)\|$ rather than x, assuming $\mathcal{W} = [1\,,\,0; 0\,,\,1]$ and $n = 2$. The function $h(x)$ (in red, dashed) shown here as a function of $f_i(x)$, is derived from $\|\frac{\partial\phi}{\partial y}\|$ and represents a tighter Lipschitz bound on the Gaussian observation.

A visualization of $h(x)$ is given in Fig. 6.2. We can now use Lemma 6.1 and the manner of choosing y^i (6.19) to show the following theorem.

Theorem 6.1. *For grid scheme $\mathcal{G}$, information state $\sigma \in \Sigma$, $u \in \mathcal{U}$, and y^i chosen to satisfy* (6.19)*, the error between $V_n^*(\sigma)$ and $V_{n,\mathcal{G}}^*(\sigma)$ for any $n \in [0, N]$ satisfies*

$$V_n^*(\sigma) - V_{n,\mathcal{G}}(\sigma) \leq \delta^y \sum_{i=1}^{m} |\overline{K}_i| \left[\overline{h} \int_{K \cap I \cap A_i} \sigma(x)\, dx \right.$$
$$\left. + h \left(\int_{\{K \cap I\} \setminus A_i} (Ax + g(u))\sigma(x)\, dx \right) \right] + \frac{\epsilon}{N} \quad (6.23)$$

with h and A_i defined in Lemma 6.1, $|\overline{K}_i|$ the Lebesgue measure of set $\overline{K}_i$, and I the support of σ.

Proof. First, we define $\overline{K}$ such that $\phi(\mathcal{Y}\backslash\overline{K}; Cx, \mathcal{W}) \leq \frac{\epsilon}{N}$. Then

$$\begin{aligned}
V_n^*(\sigma) - V_{n,\mathcal{G}}^*(\sigma) = & \int_{\overline{K}} \int_{K\cap I} \alpha_{n+1}^{*(y)}(x')\phi(dy; Cx', \mathcal{W})\sigma(dx) \\
& - \sum_{i=1}^{m} \int_{K\cap I} \alpha_{n+1}^{*(y^i)}(x')\phi(\Theta(y^i); Cx', \mathcal{W})\sigma(dx) \\
& + \int_{\overline{K}^c} \int_{K\cap I} \alpha_{n+1}^{*(y)}(x')\phi(dy; Cx', \mathcal{W})\sigma(dx) \\
& - \int_{K\cap I} \alpha_{n+1}^{*(\psi_y)}(x')\phi(\overline{K}^c; Cx', \mathcal{W})\sigma(dx) \\
\leq & \int_{\overline{K}} \int_{K\cap I} \min\{\alpha_{n+1}^{*(y)}(x'), \alpha_{n+1}^{*(\theta(y))}(x')\} \\
& \times \big(\phi(y; Cx', \mathcal{W}) - \phi(\theta(y); Cx', \mathcal{W})\big)\, \sigma(dx)\, dy + \frac{\epsilon}{N}
\end{aligned}$$

with $x' = Ax + g(u)$. Since $\alpha_{n+1}(x)$ is a probability and hence bounded above by one, and applying Lemma 6.1, we obtain

$$\begin{aligned}
V_n^*(\sigma) - V_{n,\mathcal{G}}^*(\sigma) \leq & \sum_{i=1}^{m} \int_{\overline{K}_i} \int_{K\cap I} h(Ax + g(u))\|y - y^i\|_2 \sigma(dx)\, dy + \frac{\epsilon}{N} \\
\leq & \, \delta^y \sum_{i=1}^{m} |\overline{K}_i| \Bigg[\bar{h} \int_{K\cap I\cap A_i} \sigma(dx) \\
& + \int_{\{K\cap I\}\backslash A_i} h(Ax + g(u))\sigma(x)\, dx \Bigg] + \frac{\epsilon}{N}
\end{aligned}$$

Finally, recalling that h is convex, we use Jensen's inequality to take integration over σ inside the function evaluation h, which is much easier to evaluate than the original integral, completing the proof. ■

Theorem 6.1 provides a systematic scheme for generating an adaptive grid $\mathcal{G}$ of the observation space $\mathcal{Y}$. Starting with a coarse grid of rectangular cells, an upper bound on the error in each cell is calculated using the term inside the summation of (6.23). Each cell is subdivided at the midpoint of its longest edge, until either all errors are less than some tolerance, or the sum of all errors is less than some tolerance. The procedure is summarized in Algorithm 6.1.

Note that we assume both V_n^* and $V_{n,\mathcal{G}}^*$ are calculated recursively from the exact value function V_{n+1}^* in Theorem 6.1. We can extend the above theorem to express the error introduced when using $V^*_{n,\mathcal{G}}$ *for all* n, i.e. when $\alpha^{\mathcal{G}}_{y^i,u,\sigma}$ is modified from (6.18) to

$$\alpha^{\mathcal{G}}_{y^i,u,\sigma}(x) = \int_{\mathcal{X}} \alpha^{*(y^i)}_{n+1,\mathcal{G}}(x')\phi(\Theta(y^i); Cx', \mathcal{W})\tau(dx'|x,u)1_K(x) \tag{6.24}$$

and using an induction argument.

Theorem 6.2. *For grid scheme $\mathcal{G}_n$ used at time $n \in [0,N]$, the error between the exact value function $V^*_n(\sigma)$ and $V^*_{n,\mathcal{G}}(\sigma)$ calculated using α-functions (6.24) satisfies*

$$\begin{aligned} V^*_n(\sigma) - V^*_{n,\mathcal{G}}(\sigma) \leq \delta^y \sum_{j=n}^{N} \sum_{i=1}^{m_j} |\overline{K}^j_i| \Bigg[\overline{h} \int_{K\cap I\cap A^j_i} \sigma(x)\,dx \\ +h\left(\int_{\{K\cap I\}\setminus A^j_i} (Ax+g(u))\sigma(x)\,dx\right)\Bigg] + \frac{(N-n)\epsilon}{N} \end{aligned} \tag{6.25}$$

with m_j, $\overline{K}^j_i$ and A^j_i corresponding to the grid $\mathcal{G}_j$ at time j (since $\mathcal{G}_j$ may change at each time step).

Theorem 6.2 shows that the error converges to zero as $\delta \to 0$, and also that a lower bound is guaranteed, because (6.25) is always nonnegative. This follows because $\alpha^{*(y^i)}_{n+1,\mathcal{G}}$ is chosen optimally for only a subset of $\mathcal{Y}$, and for all $y \neq y^i$, $\alpha_{n+1,\mathcal{G}}$ is chosen sub-optimally.

6.3.3 Alpha-vectors

There is no exact closed form representation for the α-functions, so we use a vector approximation. The α-functions are calculated on a grid of uniformly spaced values in K, since $\alpha(x)$ is equal to zero for $x \notin K$. When performing the update calculation (6.9), the function τ (6.4) is approximated by mapping $Ax + g(u)$ to the nearest grid point x'. The error associated with approximating α-functions with α-vectors is small, and decreases with additional grid points.

With different types of representations for α and σ, we ensure the inner product $\langle \alpha, \sigma \rangle$ is well defined by implementing a numerical integration scheme with function

Algorithm 6.1 AdaptiveGrid

Input: $\overline{K}$, σ, TOL (error tolerance for each cell), $maxIT$ (maximum number of iterations)

Output: $\mathcal{G}$, decomposition of $\overline{K}$ into cells of varying size, whose error does not exceed TOL

```
$ctr = 0;$
$\mathcal{G} = \mathcal{G}^0$ an initial coarse grid
while $ctr < maxIT$ do
  for $i = 1, \dots, |\mathcal{G}|$ do
    $error(i) = calcError(\overline{K}_i, \sigma)$, according to (6.23)
    if $error(i) > TOL$ then
      $\left[\overline{K}_{i_1}, \overline{K}_{i_2}\right] = subdivide(\overline{K}_i)$, described in 6.3.2
      $\mathcal{G} = \{\mathcal{G} \backslash \overline{K}_i\} \bigcup \{\overline{K}_{i_1}, \overline{K}_{i_2}\}$
      $ctr + +$
    end if
  end for
end while
```

values of $\alpha(x)\sigma(x)$ taken along the grid points of α. For example, in $\mathbb{R}$, using a trapezoidal rule evaluated at grid points $x^1, \dots, x^M$ with grid spacing h, we have

$$\langle \alpha, \sigma \rangle = \frac{h}{2} \left[\alpha(x^1)\sigma(x^1) + \alpha(x^M)\sigma(x^M) + 2 \sum_{i=2}^{M-1} \alpha(x^i)\sigma(x^i) \right]. \tag{6.26}$$

6.3.4 Approximate Solution to Problem 1

Combining a point-based approach similar to Perseus [PVSP06] with our results on the truncated Gaussian belief state and on the adaptive gridding algorithm (Algorithm 6.1), we can now describe the overall algorithm for approximately solving Problem 6.1.

The first step is to generate a finite set of information states $\tilde{\Sigma} \subset \Sigma$. We take a naive sampling approach, first initializing $\sigma_0^{(i)} = \mathcal{N}(\mu^i, \Sigma_0)$ for a range of initial means μ^i that are of interest, and then sequentially sample observations according to σ_n^i and choose random actions, generating a sequence $\sigma_0^i, \ldots, \sigma_N^i$ for $i = 1, \ldots, M$ according to the formulas given in Proposition 6.1. Based on the set $\tilde{\Sigma}$, we implement Algorithm 6.2 (*PointBasedSafety*).

At each time step, we cycle through all $\sigma \in \tilde{\Sigma}$ and generate a grid $\mathcal{G}$ based on σ using Algorithm 6.1 (*AdaptiveGrid*, line 8). The vector $\alpha_{y^i,u,\sigma}$ is calculated based on the representative point $y^i \in \overline{K}_i$ in line 11 according to (6.18). The vector α_σ is defined in line 15 as the vector $\alpha_{u,\sigma}$ associated with the $u \in \mathcal{U}$ that maximizes the inner product between $\alpha_{u,\sigma}$ and σ (calculated as in (6.26)).

6.4 Numerical Examples

The benefits of 1) a truncated Gaussian representation of the belief state, and 2) adaptively gridding the observation space are demonstrated first on a simple benchmark temperature regulation problem [APLS08] extended from the previous chapter, and second on a more complicated anesthesia delivery system. The reduction in computation time using our method is significant as compared to an approach that assumes a discretized belief state, discretized observation space, or both (what we call the fully discretized version).

6.4.1 Temperature Regulation

We consider a two room heating system with state $x(n) = [x_1(n), x_2(n)]^T$, $x_i(n)$ being the temperature in degrees Celsius of room i at time n, so that $\mathcal{X} = \mathbb{R}^2$. The control input $u \in \mathcal{U} = \{0, 1, 2\}$ is a command that tells the heater to heat room one ($u = 1$), room two ($u = 2$), or shut off ($u = 0$). The effect of the input is in $q(u) \in \mathbb{Z}^2$, for

Algorithm 6.2 PointBasedSafety

Input: Safe set K, terminal time N, belief states $\tilde{\Sigma}$, $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{U}$, $\overline{K}$, dynamics (6.1) and measurements (6.2)

Output: Γ_0, a collection of vectors estimating V_0^*

```
$\mathcal{X}_g$ = a grid of values in $\mathcal{X} \cap K$
$n = N$
$TOL = tol$, $maxIT = max$
$\alpha = 1_K(\mathcal{X}_g)$, $\Gamma_n = \{\alpha\}$
for $n = N - 1$; $n \geq 0$; $n = n - 1$ do
    $\Gamma_n = \emptyset$
    for all $\sigma \in \tilde{\Sigma}$ do
        $\mathcal{G} = AdaptiveGrid(\overline{K}, \sigma, tol, max)$
        for all $u \in \mathcal{U}$ do
            for all $\overline{K}_i \in \mathcal{G}$ do
                calculate $\alpha_{y^i,u,\sigma}$ according to (6.18)
            end for
            $\alpha_{u,\sigma} = \sum_{i=1}^{|\mathcal{G}|} \alpha_{y^i,u,\sigma}$
        end for
        $\alpha_\sigma = \arg\max \langle \alpha_{u,\sigma}, \sigma \rangle$
        $\Gamma_n = \Gamma_n \cup \{\alpha_\sigma\}$
    end for
end for
```

which the i'th element is 1 if $u = i$.

$$\begin{bmatrix} x_1(n+1) \\ x_2(n+1) \end{bmatrix} = \begin{bmatrix} .9613 & .022 \\ .022 & .9613 \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} + \begin{bmatrix} 0.8 & 0 \\ 0 & 0.9333 \end{bmatrix} q(u) + \begin{bmatrix} .1002 \\ .1002 \end{bmatrix} \tag{6.27}$$

The state x_n is unknown to the controller, and a noisy observation

$$y(n) = x(n) + \mathbf{w}(n) \tag{6.28}$$

is available, with $\mathcal{Y} = \mathbb{R}^2$ and $\mathbf{w}(n) \sim \mathcal{N}(0, [0.5\ 0; 0\ 0.5])$.

	Adaptive Grid		Uniform Grid	Fully Discretized
	$TOL = 0.05$	$TOL = 0.01$	0.5 Spacing	(σ as vector)
Comp. time (s)	2949.7	7029.5	8055.9	43603.8
Nbr. of obs, $\lvert\mathcal{G}\rvert$	90 (avg.)	225 (avg.)	289	289

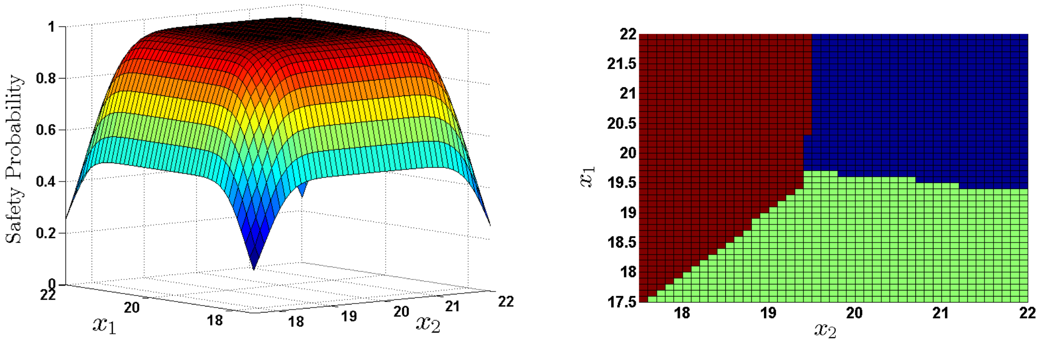
Table 6.1: Comparison of computation times. Adaptive gridding of $\mathcal{Y}$ is clearly more effective than a uniform grid spacing. Likewise, computational performance is degraded with discretization of σ.

Using the *PointBasedSafety* Algorithm, we can get an estimate of the probability that for varying initial distributions $\rho(x(0))$ the temperature stays within a desired range $K = [17.5, 22]$ over N time steps. We sample a set of 50 belief states, initialized as Gaussians with randomly selected means lying within the safe set K, $\mu_0^i \sim U[17.5, 22] \times [17.5, 22]$, and covariance fixed at $\Sigma_0 = [0.4\ 0; 0\ 0.4]$, such that $\sigma_0^i = \mathcal{N}(\mu_0^i, \Sigma_0, \mathbb{R}^2)$. Each σ is propagated using Proposition 6.1 over a period of $N = 10$ time steps.

The set $\overline{K}$ is defined as $[16, 24] \times [16, 24]$ since the probability of recording an observation associated with a state $x_n \in K$ outside of this range is negligible. The grid $\mathcal{X}_g$ used for computing the α-vectors is the set $[17.5, 22] \times [17.5, 22]$ subdivided uniformly with spacing 0.1 in each dimension.

We implement Algorithm *PointBasedSafety* first using an adaptive gridding scheme with error tolerance $TOL = .05$ in each cell, initialized with a spacing of 2 degrees in each dimension (e.g. $[16, 18; 16, 18]$ would be an initial cell), and σ represented as a truncated Gaussian. The probability of staying within K for $N = 10$ time steps as a function of the mean μ_0 from initial Gaussian distribution σ_0 is shown in Fig. 6.3a. The optimal control choice u_0 associated with each μ_0 is given in Fig. 6.3b. While for this simple system, decreasing the error tolerance to $TOL = .01$ did not result in significant improvements, for more complex systems (with more control actions to choose from) we would anticipate more sensitivity in the computed probabilistic viable sets to error tolerance.

The increase in computation time when using $TOL = .01$ rather than $TOL = .05$ is significant, because the average number of cells that must be iterated over in line 10 of Algorithm 6.2 increases from an average of 90 for $TOL = .05$ to an average of 225 for $TOL = .01$. Computation times are summarized in Table 6.1. Despite this increase in computation time, the adaptive gridding scheme is still faster than a uniformly gridded observation space, with a 0.5 grid spacing in each dimension. In the case of uniform discretization, the number of observations increases to 289, and even with only an additional 65 observations, the time to iterate over these additional observations exceeds the time it takes to compute the adaptive grid $\mathcal{G}$ for each σ. Further, we saw no distinction between the viable sets computed using an adaptive versus uniform grid on $\mathcal{Y}$.



(a) (b)

Figure 6.3: (a) The estimated probability of staying within safe set K over $N = 10$ time steps as a function of x_1, x_2 with $\sigma_0 = \mathcal{N}(\mu_0, \Sigma_0)$ and $\mu_0 = [x_1, x_2]$. (b) The optimal control input $u(0)$ as a function of μ_0. The optimal control $u(0) = 2$ is in red (i.e. heat room 2), $u(0) = 1$ in green, and $u(0) = 0$ in blue. As expected, the probability of staying within K drops towards the boundaries of K, and the optimal control is a threshhold policy that switches when the mean μ_0 falls inside certain regions. This result is computed using the *AdaptiveGridding* algorithm with $TOL = 0.05$ (computation time given in Table 6.1).

Finally, to demonstrate the additional benefit of using a truncated Gaussian representation for the information state (rather than a vector), we also ran a fully discretized version of Algorithm 6.2, in which the observations are uniformly gridded with spacing 0.5, and α and σ are stored and computed as vectors, with $\mathcal{X}_g$ a uniform grid with spacing 0.1. This is where we see the greatest increase in computation time, as shown in Table 6.1. Clearly, the combination of an adaptive grid scheme for the observation space and the truncated Gaussian representation of the information state leads to a significant decrease in computation time, and will help to make reachability and safety calculations in partially observable systems feasible across a wide range of applications.

It should be noted that we have not incorporated any techniques from advanced point-based solvers such as SARSOP [KHL08] and online planning algorithms [RPPCd08], which have led to great improvements in computation time for fully discretized POMDPs. However, such techniques should be applicable to all of the approaches we have compared, and lead to equal time improvements for the adaptive gridding and truncated Gaussian scheme as well as the fully discretized one. We intend to incorporate such techniques in future work.

6.4.2 Anesthesia Delivery

We look at a three dimensional model for anesthesia delivery [Kay12]. The three compartment pharmokinetic system determines the concentration of Propofol in different compartments of the body (the states $x(t) = [x_1(t)\, x_2(t)\, x_3(t)]^T$) given input $u(t)$, the Propofol administration rate.

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} -(k_{10}+k_{12}+k_{13}) & k_{12} & k_{13} \\ k_{21} & -k_{21} & 0 \\ k_{31} & 0 & -k_{31} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{V_1} \\ 0 \\ 0 \end{bmatrix} u(t) \qquad (6.29)$$

The parameters are selected for an 11 year old child weighing 35 kg, taken from the Paedfusor data set.

Table 6.2: Model Parameters from Paedfusor Data Set

k_{10}	k_{12}	k_{13}	k_{21}	k_{31}	V_1
0.4436	0.1140	0.0419	0.0550	0.0033	16.044

The model (6.29) is discretized using a zero-order hold with a time step of 20 seconds.

$$\begin{bmatrix} x_1(n+1) \\ x_2(n+1) \\ x_3(n+1) \end{bmatrix} = \begin{bmatrix} 0.8192 & 0.0341 & 0.0127 \\ 0.0165 & 0.9822 & 0.0001 \\ 0.001 & 0.00002 & 0.9989 \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \\ x_3(n) \end{bmatrix} + \begin{bmatrix} 0.0188 \\ 0.0002 \\ 0.00001 \end{bmatrix} u(n) \quad (6.30)$$

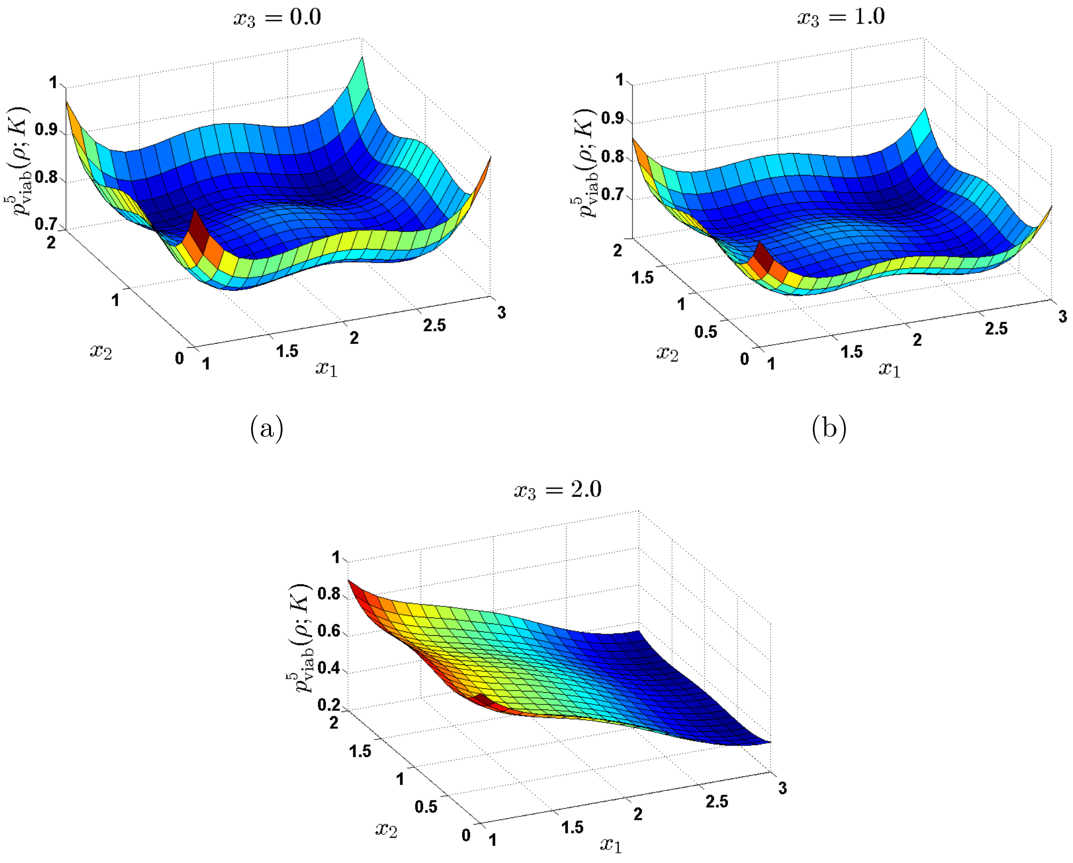
The administration rate $u(n)$ is assumed to be either a constant infusion rate of 0 mg/20 seconds, 2.5 mg/20 seconds, 5 mg/20 seconds, or 7.5 mg/20 seconds. There is also an option to deliver a bolus dose, or one concentrated injection of Propofol. The bolus dose is assumed to be 10 mg, or 30 mg/min over one 20 second time step. The control space is $\mathcal{U} = \{0, 2.5, 5, 7.5, 30\}$.

We consider the viability problem of trying to maintain propofol levels within safe limits for each compartment of the body described by model (6.30). The safe set $K = [1,3] \times [0,2] \times [0,2]$ is smaller than that considered in [Kay12] (in which $K = [1,6], [0,10], [0,10]$) in order to increase computation speed (by having smaller regions K and $\overline{K}$ to discretize) while still producing informative probabilistic viable sets.

Unlike the temperature regulation example of Section 6.4.1, we now assume the initial Gaussian distribution $\rho(x) = \phi(x; \mu_0^i, \Sigma_0, I_0)$ is also truncated. We set the support I_0 equal to K, so that we are certain that the initial propofol levels $[x_1(0), x_2(0), x_3(0)]$ lie inside the safe region. The covariance Σ_0 is a diagonal matrix with entries $[.4, .4, .25]$. The initial mean $\mu_0^i = [x_1, x_2, x_3]^T$ is sampled uniformly over K for $i = 1, \dots, 40$. We therefore sample a set of 40 σ_0^i at time zero, with $\sigma_0^i = \mathcal{N}(\mu_0^i, \Sigma_0, K)$, and continually update each σ_0^i to produce σ_n^i for $n = 1, \dots, N$.

The observations $y(n) = [y_1(n), y_2(n), y_3(n)]^T$ satisfy

$$\begin{bmatrix} y_1(n) \\ y_2(n) \\ y_3(n) \end{bmatrix} = \begin{bmatrix} x_1(n) \\ x_2(n) \\ x_3(n) \end{bmatrix} + \mathbf{w}(n)$$



(a)

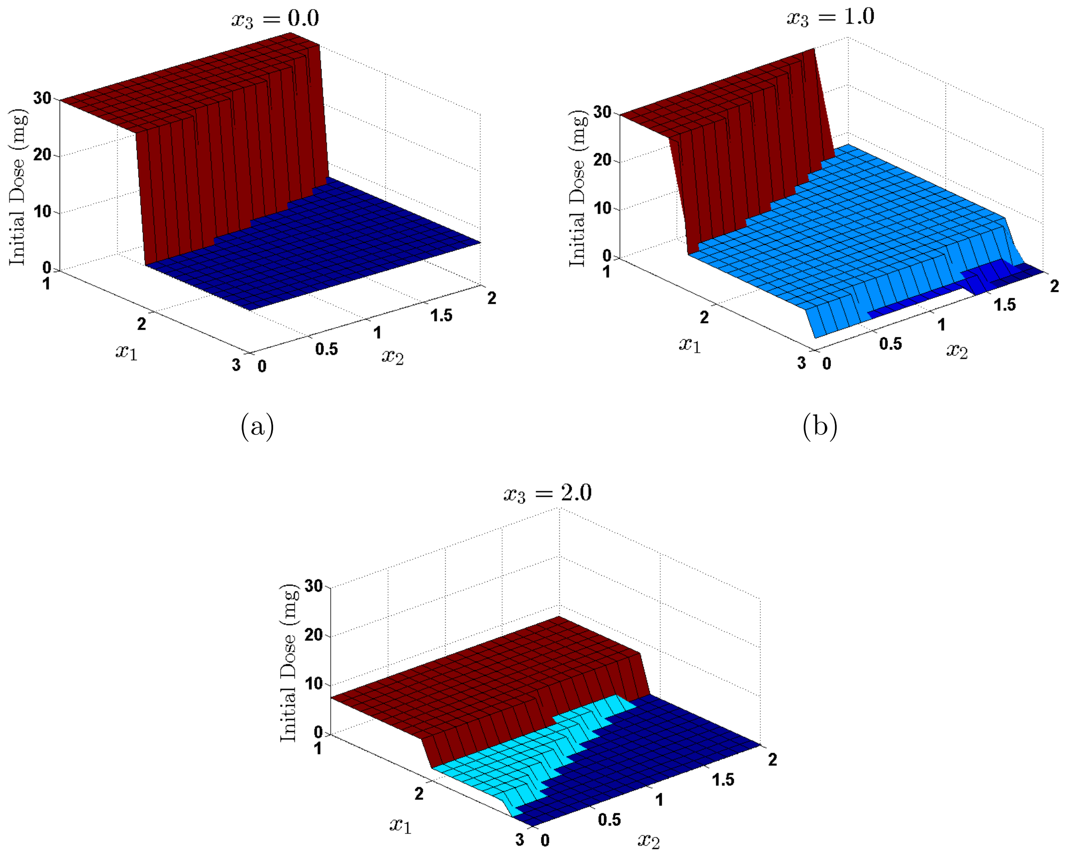
(b)

(c)

Figure 6.4: Probabilistic viable sets for initial distribution $\rho(x) = \phi(x; \mu_0, \Sigma_0, K)$ with $\mu_0 = [x_1, x_2, x_3]$. Each figure shows viability probabilities for μ_0 varying in x_1, x_2 and x_3 fixed at a) $x_3 = 0.0$, b) $x_3 = 1.0$, and c) $x_3 = 2.0$. The peaks at the boundaries of K are a result of initializing ρ as a truncated Gaussian limited to lying within K. This essentially provides additional information regarding the starting position of x_0 when μ_0 is near the boundary.

with $\mathbf{w}_n \sim \mathcal{N}(0, \mathcal{W})$ and $\mathcal{W}$ a diagonal covariance matrix with entries $[0.5, 0.5, 0.5]$, so that $\mathcal{Y} = \mathbb{R}^3$. The set $\overline{K}$ is defined as $[-1, 5] \times [-2, 4] \times [-2, 4]$.

We implement Algorithm *PointBasedSafety* using an adaptive grid scheme with tolerance $TOL = 0.05$ over $N = 5$ time steps. Fig. 6.4 shows viability probabilities over varying $\mu_0 = [x_1, x_2, x_3]$ with x_3 fixed at $x_3 = 0.0$, $x_3 = 1.0$, and $x_3 = 2.0$ in Figs. 6.4a, 6.4b, and 6.4c, respectively. The optimal initial control input $u(0)$ for varying μ_0 is shown in Fig. 6.5, again with x_3 fixed at levels $x_3 = 0.0$, $x_3 = 1.0$, and $x_3 = 2.0$.



(a)

(b)

(c)

Figure 6.5: Optimal control input $u(0)$ (initial dosage of anesthesia) for initial distribution $\rho(x) = \phi(x; \mu_0, \Sigma_0, K)$ with $\mu_0 = [x_1, x_2, x_3]$ for a) $x_3 = 0.0$, b) $x_3 = 1.0$, and c) $x_3 = 2.0$.

Initializing ρ as a truncated Gaussian restricted to set K significantly impacts the viability probabilities for μ_0 at or near the boundaries of K. Because $x(0)$ is ensured to lie inside K, our knowledge of the range of value $x(0)$ is most likely to take increases (the greatest value of the Gaussian density are restricted to a smaller region inside K). We therefore see the greatest viability probabilities at the boundaries of K, because we can make better control input decisions. There are, however, also peaks in probabilities for μ_0 lying towards the center of K, as expected.

The optimal control inputs at time zero are also intuitive. For μ_0 small, making low concentrations of propofol most likely, it is optimal to deliver a bolus dose initially.

As μ_0 increases, the initial dose decreases. For $x_3 = 2.0$, it is not optimal to deliver a bolus dose even for $x_1 = 1.0$ and $x_2 = 2.0$.

We do not compare performance for varying tolerance levels associated with the adaptive grid, nor do we compare to a fully discretized approach, as in the temperature regulation example above. We did consider various tolerances for the adaptive grid scheme, and discretization levels for the α-vectors, but found that decreasing the tolerance level or gap size for discretizing the state space did not significantly alter the results.

6.5 Summary

We have presented an algorithm for solving the partially observable safety problem for LTI systems with Gaussian measurement noise. We represent the information state as a truncated Gaussian, and adaptively grid the observation space for each σ in order to decrease computation time. In combination with a point-based approach, we present a computationally feasible method, and demonstrate its effectiveness first on a two dimensional temperature regulation problem, and second on a problem of automated anesthesia delivery. Exploiting the structure of an LTI system with Gaussian observation noise provides significant computational benefits as opposed to using the discretization or Gaussian mixture approximations of Chapter 5. However, the adaptive gridding scheme is not limited to an LTI system, and can be extended to the viability problem for a general PODTSHS $\mathcal{H}^{po}$. And again, the results presented are for the viability problem, but can be equivalently applied to a reachability or reach-avoid objective, as explained in previous chapters.

Chapter 7

Conclusion

Stochastic hybrid systems provide a modeling framework well suited to a variety of applications. The flexibility of the model, coupled with the incorporation of stochastic uncertainties, allows for insightful analysis of complex control systems. Specifically, we can use stochastic reachability analysis to generate probabilistic guarantees of safety and reachability for such systems, which may be more insightful when failure to meet the given objective (safety, reachability, etc.) is possible but unlikely.

Although stochastic reachability analysis provides an elegant formulation for verification of hybrid systems, its applicability to larger scale, realistic systems is currently limited. Computationally, the generation of either deterministic or stochastic reachable sets is a difficult task, and current methods do not scale well to higher-dimensional problems except for special cases, such as deterministic linear systems with well behaved safe or target sets.

The task of generating reachable sets in the presence of a stochastic observation process is even more difficult, and has received almost no attention. It is, however, an important problem, especially when synthesizing controllers to meet safety or reachability specifications. Many systems, such as the space docking and anesthesia examples considered in this dissertation, must use sensors to measure information about the state of the system, which is then used as an input to the controller.

The aim of this dissertation was to address both of the above areas that still need development in order to make reachability analysis a more widely applicable verification tool. We have addressed both computational concerns for reachability analysis of a perfectly observable discrete time stochastic linear system, as well as theoretical and computational concerns for reachability analysis of a partially observable discrete time stochastic hybrid system. There is, however, much more to be done, particularly in the area of computational methods for partially observable systems. We first summarize our contributions in Section 7.1, and describe some potential directions for future work in Section 7.2.

7.1 Summary of Contributions

For a linear stochastic system with additive Gaussian noise, we have presented two methods for generating reachable sets that do not require dynamic programming, and so can be applied to higher dimensional systems, in Chapter 3. The first method uses a particle approximation to estimate the reachability probability. We generate trajectories of the state using a Monte Carlo method to sample from the Gaussian noise (thus generating “particles”). Using the sample trajectories we formulate a mixed integer linear program and design a controller to maximize the number of trajectories that satisfy the reachability objective. The particle approximation method works to design both an open-loop and closed-loop controller in linear feedback form $u = Wx$. The second method exploits the linear and Gaussian nature of the dynamics to design a convex optimization problem, with the probabilistic reachability objective a chance constraint to be enforced. The parameter we optimize is the tolerance level α such that the reachability objective is satisfied with probability at least $1 - \alpha$. The convex formulation only produces an open-loop controller. Both methods were demonstrated on a spacecraft rendezvous example in four dimensions.

We then examine a DTSHS with noisy observations of the continuous and discrete states in Chapter 4. The general approach for optimal control of a partially observable system is to generate an equivalent information state that is fully known, and to solve a dynamic program over the information state instead. For an additive cost function, the information state is the distribution of the current state of the system conditioned on all past observations and actions. For the non-additive reachability cost function, this result no longer holds. We therefore derived a novel sufficient statistic that produces a perfectly observed information state for optimal control of the DTSHS to satisfy reachability specifications, which incorporates the probability that all past states have remained within the safe region (for the viability problem) in addition to the conditional distribution of the current state. We also gave reachability and reach-avoid formulations for the information state, and an information state that incorporates a change of measure, rendering the stochastic observation processes independent and identically distributed. We then provided a dynamic programming formulation in terms of the information state, and proved that the new problem over the information state is equivalent to the original reachability problem.

The dynamic programming formulation derived in Chapter 4 is important for establishing a solution framework, but is difficult to implement numerically. We therefore considered approximate reachable set computation in Chapter 5 using the information state and value function derived in Chapter 4. The similarities between a PODTSHS and a POMDP inspired the use of existing approximation techniques for optimal control of POMDPs. We first proved that the value function for the reachability problem shares the same properties (convex, piecewise-linear in the case of finite observations and control inputs) as for an additive cost POMDP, and we can therefore produce a lower bound to the value function by sampling from the information state in a process known as point-based value iteration. However, because the hybrid state and observation spaces are not finite, additional steps must be taken to approximately represent the information state and value function before using PBVI. We developed two approaches. The first is a finite state approximation to the DTSHS, which enables the information state and value function to be represented in vector form. The second approximates the information state and value function as Gaussian mixtures, by representing the indicator function over convex region K as a Gaussian mixture. In both cases we discretized the observation space to ensure the value function is piecewise-linear. We then presented the first numerical results for calculating reachable sets for a PODTSHS.

Finally, we considered the special case of a system with linear dynamics, no process noise, and Gaussian observation noise in Chapter 6. We showed that in this case the information state is a truncated Gaussian distribution, and does not need to be approximated. We still use PBVI, but developed an adaptive discretization scheme for the observations to generate fewer finite observations while minimizing the error from the approximation. We presented an algorithm for choosing the observations, and for approximating the reachability probabilities. The simple form of the information state and the selection of the finite observations enabled much faster reachable set computation as compared to the approaches presented in Chapter 5 for a general PODTSHS. We were then able to compute stochastic viable sets for an automated anesthesia delivery example.

7.2 Future Directions

The work we have presented leaves many opportunities for further research. Particularly for reachability analysis of a PODTSHS, where so little work has been done, there are many possible directions to take. We describe below some of the directions for future research that we believe to be the most interesting and promising.

First, reachable set computation for perfectly observable systems without dynamic programming must be further explored for stochastic reachability analysis to be applicable to higher dimensional systems. Ideally, we would like to find results for linear systems with Gaussian noise that are equivalent to the ellipsoidal/zonotope/support vector representations of the deterministic reachable set for linear systems. It may be possible to propagate the $1 - \alpha$ probabilistic reachable set in the same fashion as the deterministic reachable set for linear systems, described in [KV07], [Gir05], and [GG10]. Short of that, improving upon the single-stage stochastic optimization formulations presented in Chapter 3 may be possible. The advantage of the convex formulation over the particle approximation is that it guarantees a lower bound to the reachable set, however we were unable to implement a feedback controller as for the particle approximation. We may be able to use model predictive control to simulate feedback and to get a better estimate of the reachable set while still solving a convex optimization problem. Another possibility is to use stochastic branch and bound methods [NPR98] to generate open loop controllers, which may be faster than the particle or convex approaches, although this requires further exploration.

There are also many possibilities for improvements to the approximate computation of reachable sets for a PODTSHS. We implemented a naive PBVI algorithm that sampled information states randomly and required backing up every information state at every time step. Other PBVI algorithms have been developed that significantly reduce computation time, such as [KHL08] and [SS04], by sampling information states in a more systematic fashion and producing iterative approximations to the value function. However, these methods are defined for optimization over an infinite horizon, so to calculate reachability probabilities over a finite horizon they will need to be modified, if possible. Another possibility is to use an online POMDP planning algorithm [RPPCd08], which can be done over finite horizons, and may prove faster than PBVI methods in some cases.

Whichever existing POMDP algorithm we use, its speed will be limited by the size of the problem. We would therefore like to find smaller, simpler representations of any system we wish to analyze. The further study of abstractions for a PODTSHS may be helpful in generating equivalent, simpler systems. For example, the adaptive gridding procedure presented in Chapter 6 is not limited to a linear system without process noise. We should therefore be able to use a similar gridding procedure for both the continuous state space $\mathcal{X}$ and observation space $\mathcal{Y}^x$ to reduce the size of the finite state approximation given in Chapter 5. There may also be other classes of systems that admit a simple expression for the information state, as is the case for a linear system without process noise and with Gaussian observation noise.

As demonstrated in Chapter 6, the truncated Gaussian representation of the information state leads to significant improvements in computation time to generate the reachable sets. However, there are still some details that require attention to improve its applicability. For our numerical examples we only used safe sets represented as hyper-rectangles, i.e. the safe set K is represented as $K = \{x_i : a_i \leq x_i \leq b_i, \forall i = 1, \ldots, n\}$ for an n-dimensional system, because the support of the Gaussian maintains this shape and we only need to update the values a_i and b_i at each time step. However, we give a general expression for the support of the truncated Gaussian at each time step without the assumption that K is a hyper-rectangle, which turns out to be the same expression as for the viability kernel as described in [MKM+13]. We should therefore be able to update the support using the algorithm presented in [MKM+13].

The next concern is with the integration of $\alpha(x)$ multiplied by $\sigma(x)$. We only store the value of $\alpha(x)$ at a finite set of points x^i, and must numerically integrate $\int_I \alpha(x)\sigma(x)dx$ over the support I of $\sigma(x)$. In low dimensions, and for I a hyper-rectangle, we can easily and fairly accurately approximate the integral using, e.g., the trapezoid or Simpson's rule for integration at the points x^i. In higher dimensions, and with I a more complex region, this is no longer possible. We are currently investigating ways to approximate the integral over an arbitrary convex region I using Gaussian quadrature methods. Once an adequate numerical integration scheme is found, we plan on applying the method of Chapter 6 to the space docking problem of Chapter 3 with the Gaussian process noise removed, and with the inclusion of Gaussian measurements. Finally, we would like to explore the possibility of extending the truncated Gaussian representation to piecewise-linear systems that allow switching between linear dynamics dependent on a discrete mode.

Appendix A

Proofs of Theorems from Chapter 4

A.1 Proof of Theorem 4.1

In order to prove the Theorem 4.1, we first define some notation and give some intermediate results. The following proofs are based on those in [BS96], Chapters 6 and 11. To facilitate the connection between these proofs and those appearing in [BS96] we first reformulate the recursion (4.26) as a minimization

$$\sup_{\tilde{\pi}} V_0^{\tilde{\pi}}(\sigma_0) = -\inf_{\tilde{\pi}} -V_0^{\tilde{\pi}} = -\inf_{\tilde{\pi}} J_0^{\tilde{\pi}}$$

Let $J_n^{\tilde{\pi}}(\sigma) = -V_n^{\tilde{\pi}}(\sigma)$ and $\tilde{\Pi} = \{\tilde{\pi} = (\tilde{\mu}_0, \tilde{\mu}_1, \dots) : \tilde{\mu}_n(\sigma_n) \in \mathcal{U}\,\forall\, n \in [0, N]\}$. The recursion for $J_n^*(\sigma)$ is identical to that of $V_n^*(\sigma)$ in (4.26) except that $J_N^*(\sigma) = -\langle \sigma, 1_K \rangle$.

Next we define the operators

$$\begin{aligned} H_{\tilde{\mu}}[J] &= \mathbb{E}^{\tilde{\pi}}\left[J(\Phi_{y,\tilde{\mu}(\sigma)}\sigma)\right] \\ &= \int_K \int_{\mathcal{S}} \int_{\mathcal{Y}} J(\Phi_{y,\tilde{\mu}(\sigma)})\tau_y(dy|s', \tilde{\mu}(\sigma))\tau_s(ds'|s, \tilde{\mu}(\sigma))\sigma(ds) \\ H[J] &= \inf_{\tilde{\mu}(\sigma) \in \mathcal{U}} H_{\tilde{\mu}}[J] \end{aligned}$$

The operator $H_{\tilde{\mu}}[J]$ preserves the linearity and boundedness of value function $J(\sigma)$ for all σ in L^1, which can be seen from a straightforward induction argument. Because $J_N^{\tilde{\pi}}(\sigma)$ is a bounded linear functional, this then implies that $J_n^{\tilde{\pi}}(\sigma)$ is a bounded linear functional for all $n = 0, \ldots, N$ and for all σ in L^1.

Lemma A.1. *For all bounded linear functionals J, $\sigma \in L^1$, $\tilde{\pi} \in \tilde{\Pi}$, and $r \in \mathbb{R}^+$*

$$H_{\tilde{\mu}}[J] \leq H_{\tilde{\mu}}[J + r] \leq H_{\tilde{\mu}}[J] + r$$

Proof. Because $J \leq J + r$ when $r \geq 0$, we get the following:

$$\begin{aligned}
H_{\tilde{\mu}}[J] &= \int_K \int_{\mathcal{S}} \int_{\mathcal{Y}} J(\Phi_{y,\tilde{\mu}(\sigma)}) \tau_y(dy|s', \tilde{\mu}(\sigma)) \tau_s(ds'|s, \tilde{\mu}(\sigma)) \sigma(ds) \\
&\leq \int_K \int_{\mathcal{S}} \int_{\mathcal{Y}} J(\Phi_{y,\tilde{\mu}(\sigma)} + r) \tau_y(dy|s', \tilde{\mu}(\sigma)) \tau_s(ds'|s, \tilde{\mu}(\sigma)) \sigma(ds) \\
&\leq H_{\tilde{\mu}}[J + r] \\
&\leq \int_K \int_{\mathcal{S}} \int_{\mathcal{Y}} J(\Phi_{y,\tilde{\mu}(\sigma)}) \tau_y(dy|s', \tilde{\mu}(\sigma)) \tau_s(ds'|s, \tilde{\mu}(\sigma)) \sigma(ds) + r \\
&\leq H_{\tilde{\mu}}[J] + r
\end{aligned}$$

■

Proposition A.1. *For any $M \in \mathbb{N}$, where $J_0^*(\sigma) = \inf_{\tilde{\pi} \in \tilde{\Pi}} J_0^{\tilde{\pi}}(\sigma)$,*

$$J_0^*(\sigma) = H^M[J_M^*](\sigma)$$

Further, for any $\epsilon > 0$ there exists an M-stage ϵ-optimal policy $\tilde{\pi}_\epsilon$, defined as

$$J_0^* \leq J_0^{\tilde{\pi}_\epsilon} \leq J_0^* + \epsilon$$

Proof. By backwards induction. For $M = N$,

$$J_N^*(\sigma) = H^0[J_N^*](\sigma)$$

because $H^0[J] = J$. Also, because J_N does not depend on a control input, $J_N^*(\sigma) = J_N^{\tilde{\pi}}(\sigma)$ for any policy $\tilde{\pi} \in \tilde{\Pi}$. Therefore, for any $\epsilon > 0$, $J_N^{\tilde{\pi}_\epsilon}(\sigma) = J_N^*(\sigma) \leq J_N^*(\sigma) + \epsilon$.

Assume for $M = n+1$ that $J^*_{n+1}(\sigma) = H^{N-n-1}[J^*_N](\sigma)$ and that for all $\epsilon > 0$ there exists a policy $\tilde{\pi}_\epsilon$ such that $J^{\tilde{\pi}_\epsilon}_{n+1}(\sigma) \le J^*_{n+1}(\sigma) + \epsilon$. Then by Lemma A.1, for any $\tilde{\mu}$ an element of $\tilde{\pi}$ such that $\tilde{\pi} \in \tilde{\Pi}$,

$$H_{\tilde{\mu}}[J^{\tilde{\pi}_\epsilon}_{n+1}] \le H_{\tilde{\mu}}[J^*_{n+1} + \epsilon] \le H_{\tilde{\mu}}[J^*_{n+1}] + \epsilon.$$

By aggregating $\tilde{\mu}$ with the control policy $\tilde{\pi}_\epsilon$ to get $\hat{\tilde{\pi}}_\epsilon = (\tilde{\mu}, \tilde{\pi}_\epsilon)$, we then have

$$\begin{aligned}\inf_{\tilde{\pi}} J^{\tilde{\pi}}_n(\sigma) \le J^{\hat{\tilde{\pi}}_\epsilon}_n(\sigma) &= H_{\tilde{\mu}}[J^{\tilde{\pi}_\epsilon}_{n+1}] \\ &\le H_{\tilde{\mu}}[J^*_n] + \epsilon.\end{aligned}$$

Since the above holds for *any* $\tilde{\mu} \in \tilde{\Pi}$,

$$\begin{aligned}\inf_{\tilde{\pi}} J^{\tilde{\pi}}_n(\sigma) &\le H[J^*_{n+1}](\sigma) \\ &= H[H^{N-n-1}[J^*_N]](\sigma) = H^{N-n}[J^*_N](\sigma).\end{aligned}$$

By definition $H^{N-n}[J^*_N](\sigma) \le J^*_n(\sigma)$, hence $H^{N-n}[J^*_N](\sigma) = J^*_n(\sigma)$.

Next, by the induction argument, for any $\hat{\epsilon} > 0$, define $\hat{\tilde{\pi}}$ so that

$$J^{\hat{\tilde{\pi}}}_{n+1}(\sigma) \le J^*_{n+1}(\sigma) + \frac{\hat{\epsilon}}{2}$$

Define $\hat{\tilde{\mu}} \in \tilde{\Pi}$ so that

$$H_{\hat{\tilde{\mu}}}[J^*_{k+1}] \le H[J^*_{k+1}] + \frac{\hat{\epsilon}}{2}$$

Define $\hat{\tilde{\pi}}_\epsilon = (\hat{\tilde{\mu}}, \hat{\tilde{\pi}})$. Then

$$\begin{aligned}J^{\hat{\tilde{\pi}}_\epsilon}_n = H_{\hat{\tilde{\mu}}}[J^{\hat{\tilde{\pi}}}_{n+1}] &\le H_{\hat{\tilde{\mu}}}[J^*_{n+1}] + \frac{\hat{\epsilon}}{2} \\ &\le H[J^*_{n+1}] + \frac{\hat{\epsilon}}{2} + \frac{\hat{\epsilon}}{2} \\ &= J^*_n + \hat{\epsilon}\end{aligned}$$

It follows from induction that $J^*_0 \le J^{\tilde{\pi}_\epsilon}_0 \le J^*_0 + \epsilon$ for any M. ■

We also use the result from [BS96] on the existence of a uniformly N-stage optimal policy $\tilde{\pi}^* = (\tilde{\mu}^*_0, \tilde{\mu}^*_1, ...)$, which we give without proof (see Ch. 6), since the proof does not change in our context.

Proposition A.2. *A policy is uniformly N-stage optimal if and only if $H_{\tilde{\mu}_n^*}[H^{N-n-1}[J_N^*]] = H^{N-n}[J_N^*]$ for all $n = 0, \ldots, N$, and this policy exists if and only if the infimum of*

$$H^{N-n}[J_N^*] = \inf_{u \in \mathcal{U}} H_u \left[H^{N-n-1}[J_N^*] \right]$$

is attained for all $\sigma \in L^1$ and $n = 0, \ldots, N$. A sufficient condition for the infimum to be attained is that

$$U_n(\sigma, \lambda) = \left\{ u \in \mathcal{U} : H_u \left[H^{N-n-1}[J_N^*] \right] \leq \lambda \right\}$$

is compact for all $\sigma \in L^1$, $\lambda \in \mathbb{R}$, and $n = 0, \ldots, N$.

We can now use the above results to prove Theorem 4.1.

Proof. (Of Theorem 4.1)

Substituting $V_n^* = -J_n^*$, it is clear that Prop. A.1 validates the dynamic programming algorithm (4.26), and proves the existence of at least an ϵ-optimal policy, and so the first part of Theorem 4.1 is proved. Finally, using Prop. A.2, because $\mathcal{U}$ is defined as a compact (i.e. closed and bounded) Borel set, and J_n (and so V_n) is bounded for all $\sigma \in L^1$ and for each $u \in \mathcal{U}$, then there exists some $u \in \mathcal{U}$ such that the infimum in $\inf_{u \in \mathcal{U}} H_u \left[H^{N-n-1}[J_N^*] \right] = \inf_{u \in \mathcal{U}} \mathbb{E}[J_n(\Phi_{y,u}\sigma)]$ is attained for all n (and likewise the supremum of V_n is achieved for all n). Therefore, for (4.26), there always exists an optimal policy $\tilde{\pi}^*$ given by (4.27). ■

A.2 Proof of Theorem 4.2

Proof. (Of Theorem 4.2) For a vector $\mathbf{u} = [u_0, u_1, \ldots u_{N-1}]$ with each $u_n \in \mathcal{U}$, we have by definition that

$$r_K(\mathbf{u}) = \overline{r}_K(\mathbf{u}) \quad \forall\, \mathbf{u} \in \mathcal{U}_N$$

Since $\sigma_n = \eta_n(\rho, i_n)$, the control policy $\tilde{\pi} = (\tilde{\mu}_0(\sigma_0), \tilde{\mu}_1(\sigma_1), \ldots)$ can be rewritten as a function of the information vector i_n, where $\tilde{\mu}_n(\sigma_n) = \tilde{\mu}_n(\eta_n(\rho, i_n)) = \mu_n(i_n)$. Then by defining the policy π in terms of μ, we obtain $p^N_{\text{viab}}(\rho; K) = \tilde{p}^N_{\text{viab}}(\rho; K)$ for *any* $\tilde{\pi} \in \tilde{\Pi}$. If $\tilde{\pi}^*$ is optimal for $\tilde{p}^N_{\text{viab}}(\rho; K)$, it then must be optimal for $p^N_{\text{viab}}(\rho; K)$ as well, and further,

$$p^N_{\text{viab}}(\rho; K) = \tilde{p}^N_{\text{viab}}(\rho; K)$$

■

Appendix B

Gaussian Mixture Results from Chapter 5

We provide constructive proofs of Lemmas 5.6 and 5.7 from Chapter 5. Lemma 5.6 leads to a recursive expression for $\sigma_{n+1,g}$ in terms of the weights, means, and covariances characterizing $\sigma_{n,g}$. Similarly, Lemma 5.7 leads to a recursive expression for $\alpha^{g}_{y,u,\sigma}$ in terms of the weights, means, and covariances characterizing $\alpha^{*(y)}_{n+1,g}$. We first prove Lemma 5.6, show that $\sigma_{n,g}$ can therefore be represented by a Gaussian mixture for all $n \in [0, N]$, and give the recursive expression for $\sigma_{n+1,g}$ given $\sigma_{n,g}$ in Section B.1. The process is repeated for the α-functions in Section B.2.

Before providing proofs, we give some identities for Gaussian distributions. First, the product of two Gaussian densities is again a Gaussian density, up to a constant factor.

$$\phi(x;\mu_1,\Sigma_1)\phi(x;\mu_2,\Sigma_2) = \phi(\mu_1;\mu_2,\Sigma_1+\Sigma_2)\phi(x;\tilde{\mu},\tilde{\Sigma}) \tag{B.1}$$

with

$$\begin{aligned}\tilde{\mu} &= \tilde{\Sigma}\left(\Sigma_1^{-1}\mu_1 + \Sigma_2^{-1}\mu_2\right)\\ \tilde{\Sigma} &= \left(\Sigma_1^{-1}+\Sigma_2^{-1}\right)^{-1}\end{aligned} \tag{B.2}$$

Second, for invertible matrix A, constant b, and variables x and y,

$$\phi(y; Ax+b,\Sigma) = |A^{-1}|\phi(x; A^{-1}(y-b), A^{-1}\Sigma A^{-T}). \tag{B.3}$$

Both identities are easily shown directly. Also recall the radial basis function approximation $1_K(s)$ is given by

$$1_K(x,q) \approx \sum_{i=1}^{I_q} w_i^I(q)\phi(x;\mu_i^I,\Sigma_i^I) \tag{B.4}$$

B.1 Information State as a Gaussian Mixture

Proof. (Of Lemma 5.6) By construction. Given $\sigma_{n,g}(x,q) = \sum_{l=1}^{L} w_{l,n}^{\sigma}(q)\phi(x;\mu_{l,n}^{\sigma}(q),\Sigma_{l,n}^{\sigma}(q))$, observation $y \in \mathcal{Y}$, and control input $u \in \mathcal{U}$, the operator $\Phi_{y,u}^{g}\sigma_{n,g}$ produces

$$\begin{aligned}\Phi_{y,u}^{g}\sigma_{n,g} = \frac{1}{\mathbb{P}(y|\sigma_{n,g},u)}&\tau_y(y|s',u)\\ &\times \sum_{q\in\mathcal{Q}}\int_{\mathcal{X}}\left[\sum_{i=1}^{I_q} w_i^I(q)\phi_i^I(x)\right]\tau_s(s'|s,u)\sigma_{n,g}(s)\,dx.\end{aligned} \tag{B.5}$$

The normalizing factor $\mathbb{P}(y|\sigma_{n,g},u)$ is removed, because the Gaussian mixture can be normalized independently be rescaling all of the weights to sum to one. Replacing $\sigma_{n,g}$ by its Gaussian mixture representation in (B.5), and expanding τ_y and τ_s, gives

$$\begin{aligned}\left(\Phi_{y,u}^{g}\sigma_{n,g}\right)(x',q') = T_{y^q}(y^q|q',u)\phi(y^x;x',\mathcal{W})&\sum_{q\in\mathcal{Q}}\int_{\mathcal{X}}\left[\sum_{i=1}^{I_q} w_i^I(q)\phi_i^I(x)\right]\\ &\times T_q(q'|q,u)\phi(x';Ax+g(q,u,q'),\mathcal{V})\\ &\times\left[\sum_{l=1}^{L} w_{l,n}^{\sigma}(q)\phi(x;\mu_{l,n}^{\sigma}(q),\Sigma_{l,n}^{\sigma}(q))\right]\end{aligned}$$

$$= \sum_{q=1}^{N_q} \sum_{l=1}^{L} \sum_{i=1}^{I_q} w_i^I(q) w_{m,n}^\sigma(q) T_{y^q}(y^q|q', u) T_q(q'|q,u) \phi(x'; y^x; \mathcal{W})$$

$$\times \int_{\mathcal{X}} \phi(x; \mu_i^I(q), \Sigma_i^I(q)) \phi(x; \mu_{l,n}^\sigma(q), \Sigma_{l,n}^\sigma(q))$$

$$\times |A^{-1}| \phi(x; A^{-1}(x' - g(q,u,q')), A^{-1}\mathcal{V}A^{-T})\, dx \tag{B.6}$$

$$= \sum_{q=1}^{N_q} \sum_{l=1}^{L} \sum_{i=1}^{I_q} |A^{-1}| w_i^I(q) w_{m,n}^\sigma(q) T_{y^q}(y^q|q', u) T_q(q'|q,u) \phi(x'; y^x; \mathcal{W})$$

$$\times \phi(\mu_i^I(q); \mu_{l,n}^\sigma(q), \Sigma_i^I(q) + \Sigma_{l,n}^\sigma(q)) \int_{\mathcal{X}} \phi(x; \tilde{\mu}, \tilde{\Sigma})$$

$$\times \phi(x; A^{-1}(x' - g(q,u,q')), A^{-1}\mathcal{V}A^{-T})\, dx \tag{B.7}$$

$$= \sum_{q=1}^{N_q} \sum_{l=1}^{L} \sum_{i=1}^{I_q} |A^{-1}| w_i^I(q) w_{m,n}^\sigma(q) T_{y^q}(y^q|q', u) T_q(q'|q,u) \phi(x'; y^x; \mathcal{W})$$

$$\times \phi(\mu_i^I(q); \mu_{l,n}^\sigma(q), \Sigma_i^I(q) + \Sigma_{l,n}^\sigma(q)) \phi(A^{-1}(x' - g(q,u,q'); \tilde{\mu}, \tilde{\Sigma})) \tag{B.8}$$

Line (B.6) follows from (B.3), line (B.7) from combining $\phi(x; \mu_i^I(q), \Sigma_i^I(q))$ and $\phi(x; \mu_{l,n}^\sigma(q), \Sigma_{l,n}^\sigma(q))$ according to (B.1), and (B.8) from a second application of (B.1) and setting the integral of a Gaussian density over $\mathcal{X}$ equal to one.

A final application of (B.1) and (B.3) gives

$$\left(\Phi_{y,u}^g \sigma_{n,g}\right)(x', q') = \sum_{q=1}^{N_q} \sum_{l=1}^{L} \sum_{i=1}^{I_q} w_i^I(q) w_{m,n}^\sigma(q) T_{y^q}(y^q|q', u) T_q(q'|q,u)$$

$$\times \phi(\mu_i^I(q); \mu_{l,n}^\sigma(q), \Sigma_i^I(q) + \Sigma_{l,n}^\sigma(q))$$

$$\times \phi(y^x; A\tilde{\mu} + g(q,u,q'), \mathcal{W} + \mathcal{V} + A\tilde{\Sigma}A^T)$$

$$\times \phi(x; \mu_{q,l,i,n+1}^\sigma(q'), \Sigma_{q,l,i,n+1}^\sigma(q'))$$

which is again a Gaussian mixture with $N_q L I_q$ components. ■

Using Lemma 5.6, we can make a stronger statement regarding the information states.

Theorem B.1. *Given RBF approximation* (B.4) *to* $1_K(s)$ *and operator* $\Phi_{y,u}^g$ (B.5), *the information state* $\sigma_{n,g}$ *satisfy the following.*

1. *The information states $\sigma_{n,g}(s)$ are Gaussian mixtures for all $n \in [0, N]$, for any sequence of control inputs $u_0, \ldots, u_{N-1}$, $u_n \in \mathcal{U}$, and any sequence of observations $y_1, \ldots, y_N$, $y_n \in \mathcal{Y}$.*

2. *For $\sigma_{n,g}(x,q) = \sum_{l=1}^{L} w^{\sigma}_{l,n}(q)\phi(x; \mu^{\sigma}_{l,n}(q), \Sigma^{\sigma}_{l,n}(q))$, observation $y \in \mathcal{Y}$ and control input $u \in \mathcal{U}$, $\sigma_{n+1,g}(x', q')$ is given by*

$$\sigma_{n+1,g}(x', q') = \sum_{q=1}^{N_q} \sum_{l=1}^{L} \sum_{i=1}^{I_q} w^{\sigma}_{q,l,i,n+1}(q')\phi(x'; \mu^{\sigma}_{q,l,i,n+1}(q'), \Sigma^{\sigma}_{q,l,i,n+1}(q')),$$

with

$$\begin{aligned} w^{\sigma}_{q,l,i,n+1}(q') &= w^{I}_{i}(q) w^{\sigma}_{l,n}(q) T_{y^q}(y^q|q', u) T_q(q'|q, u) \\ &\qquad \times \phi(\mu^{I}_{i}(q); \mu^{\sigma}_{l,n}(q), \Sigma^{I}_{i}(q) + \Sigma^{\sigma}_{l,n}(q)) \\ &\qquad \times \phi(y^x; A\tilde{\mu} + g(q,u,q'), \mathcal{W} + \mathcal{V} + A\tilde{\Sigma}A^T), \\ \mu^{\sigma}_{q,l,i,n+1}(q') &= \Sigma^{\sigma}_{q,l,i,n+1}(q') \left[\mathcal{W}^{-1} y^x + \left(A\tilde{\Sigma}A^T + \mathcal{V}\right)^{-1} (A\tilde{\mu} + g(q,u,q'))\right], \\ \Sigma^{\sigma}_{q,l,i,n+1}(q') &= \left[\mathcal{W}^{-1} + \left(A\tilde{\Sigma}A^T + \mathcal{V}\right)^{-1}\right]^{-1}, \end{aligned}$$

and

$$\begin{aligned} \tilde{\mu} &= \tilde{\Sigma} \left[\left(\Sigma^{I}_{i}(q)\right)^{-1} \mu^{I}_{i}(q) + \left(\Sigma^{\sigma}_{l,n}(q)\right)^{-1} \mu^{\sigma}_{l,n}(q)\right], \\ \tilde{\Sigma} &= \left[\left(\Sigma^{I}_{i}(q)\right)^{-1} + \left(\Sigma^{\sigma}_{l,n}(q)\right)^{-1}\right]. \end{aligned}$$

Proof. Part one of the theorem follows by induction. At time $n = 0$, $\sigma_{0,g}(x,q) = \rho(x,q)$ and $\rho(x,q) = Q_0(q)\phi(x; \mu^{\sigma}_0, \Sigma^{\sigma}_0)$ by definition (see Section 5.3 of Chapter 5). The induction argument then follows directly from Lemma 5.6.

Part two of the theorem follows directly from the derivation provided in the proof of Lemma 5.6. ■

B.2 Alpha-Function as a Gaussian Mixture

Proof. (Of Lemma 5.7) By construction. Given $\alpha^{*(y)}_{n+1,g}(x', q') =$

$\sum_{m=1}^{M} w^{\alpha,y}_{m,n+1}(q',y)\phi(x';\mu^{\alpha,y}_{m,n+1}(q',y),\Sigma^{\alpha,y}_{m,n+1}(q',y))$ for some observation $y \in \mathcal{Y}$, then $\alpha^g_{y,u,\sigma}(x,q)$ for control input $u \in \mathcal{U}$ is written

$$\alpha^g_{y,u,\sigma}(x,q) = \sum_{q'\in\mathcal{Q}} \int_{\mathcal{X}} \alpha^{*(y)}_{n+1,g}(s')\tau_y(y|s',u)\tau_s(s'|s,u)\,dx' \left[\sum_{i=1}^{I_q} w^I_i(q)\phi^I_i(x)\right] \tag{B.9}$$

$$\begin{aligned}&= \sum_{q'\in\mathcal{Q}} \int_{\mathcal{X}} \left[\sum_{m=1}^{M} w^{\alpha,y}_{m,n+1}(q')\phi(x';\mu^{\alpha,y}_{m,n+1}(q'),\Sigma^{\alpha,y}_{m,n+1}(q'))\right] \\ &\qquad \times T_{y^q}(y^q|q',u)\phi(y^x;x',\mathcal{W})T_q(q'|q,u) \\ &\qquad\quad \times \phi(x';Ax+g(q,u,q'),\mathcal{V})\,dx' \left[\sum_{i=1}^{I_q} w^I_i(q)\phi^I_i(x)\right]\end{aligned} \tag{B.10}$$

$$\begin{aligned}&= \sum_{q'=1}^{N_q}\sum_{m=1}^{M}\sum_{i=1}^{I_q} w^{\alpha,y}_{m,n+1}(q')w^I_i(q)T_{y^q}(y^q|q',u)T_q(q'|q,u) \\ &\qquad \times \phi(x;\mu^I_i(q),\Sigma^I_i(q))\phi(y^x;\mu^{\alpha,y}_{m,n+1}(q'),\mathcal{W}+\Sigma^{\alpha,y}_{m,n+1}(q')) \\ &\qquad\quad \times \int_{\mathcal{X}} \phi(x';\hat{\mu},\hat{\Sigma})\phi(x';Ax+g(q,u,q'),\mathcal{V})\,dx'\end{aligned} \tag{B.11}$$

$$\begin{aligned}&= \sum_{q'=1}^{N_q}\sum_{m=1}^{M}\sum_{i=1}^{I_q} w^{\alpha,y}_{m,n+1}(q')w^I_i(q)T_{y^q}(y^q|q',u)T_q(q'|q,u) \\ &\qquad \times \phi(x;\mu^I_i(q),\Sigma^I_i(q))\phi(y^x;\mu^{\alpha,y}_{m,n+1}(q'),\mathcal{W}+\Sigma^{\alpha,y}_{m,n+1}(q')) \\ &\qquad\quad \times |A^{-1}|\phi(x;A^{-1}(\hat{\mu}-g(q,u,q')),A^{-1}(\hat{\Sigma}+\mathcal{V})A^{-T})\end{aligned} \tag{B.12}$$

$$\begin{aligned}&= \sum_{q'=1}^{N_q}\sum_{m=1}^{M}\sum_{i=1}^{I_q} w^{\alpha,y}_{m,n+1}(q')w^I_i(q)T_{y^q}(y^q|q',u)T_q(q'|q,u) \\ &\qquad \times \phi(\mu^I_i(q);A^{-1}(\hat{\mu}-g(q,u,q')),\Sigma^I_i(q)+A^{-1}(\hat{\Sigma}+\mathcal{V})A^{-T}) \\ &\qquad\quad \times \phi(y^x;\mu^{\alpha,y}_{m,n+1}(q',y),\mathcal{W}+\Sigma^{\alpha,y}_{m,n+1}(q',y)) \\ &\qquad\qquad \times \phi(x;\mu^{\alpha,y}_{q',m,i,n}(q),\Sigma^{\alpha,y}_{q',m,i,n}(q)\end{aligned} \tag{B.13}$$

Line (B.11) follows from one application of (B.1), line (B.12) from (B.3) and another application of (B.1), and a final product of Gaussian densities gives (B.13). Hence $\alpha^g_{y,u,\sigma}$ is a Gaussian mixture with $N_q M I_q$ components. ■

Lemma 5.7 leads to the following theorem concerning the α-functions as Gaussian mixtures. As expressed in Chapter 5, the α-functions have an infinite number of components when $\mathcal{Y}$ is not finite. In practice, the Gaussian mixture representation is therefore only feasible for finite $\mathcal{Y}$. We write the expression for $\alpha_{n,g}$ using a summation over $\mathcal{Y}$, and assume that the number of components for each $\alpha_{n+1,g}$ is finite, but stress that this will not be the case unless we make the additional assumption that $\mathcal{Y}$ is finite, as discussed in Section 5.6 of Chapter 5.

Theorem B.2. *Given RBF approximation* (B.4) *to* $1_K(s)$ *and the expression for* $\alpha^g_{y,u,\sigma}$ (B.9), *the α-functions satisfy the following.*

1. *The α-functions $\alpha_{n,g}(s)$ are Gaussian mixtures for all $n \in [0, N]$, for any sequence of control inputs $u_0, \ldots, u_{N-1}$, $u_n \in \mathcal{U}$.*

2. *For* $\alpha^{*(y)}_{n+1,g}(x', q') = \sum_{m=1}^{M} w^{\alpha,y}_{m,n}(q')\phi(x'; \mu^{\alpha,y}_{m,n}(q'), \Sigma^{\alpha,y}_{m,n}(q'))$ *for all indices* $*(y)$, $y \in \mathcal{Y}$, *and control inputs* $u \in \mathcal{U}$, $\alpha_{n,g}(x, q)$ *is given by*

$$\alpha_{n,g}(x,q) = \sum_{y \in \mathcal{Y}} \sum_{q=1}^{N_q} \sum_{m=1}^{M} \sum_{i=1}^{I_q} w^{\sigma}_{y,q',m,i,n+1}(q)\phi(x; \mu^{\alpha}_{y,q',m,i,n+1}(q), \Sigma^{\alpha}_{y,q',m,i,n+1}(q)),$$

with

$$\begin{aligned} w^{\alpha}_{y,q,l,i,n+1}(q) = {} & w^{\alpha,y}_{m,n+1}(q') w^{I}_{i}(q) T_{y^q}(y^q|q',u) T_q(q'|q,u) \\ & \times \phi(\mu^I_i(q); A^{-1}(\hat{\mu} - g(q,u,q')), \Sigma^I_i(q) + A^{-1}(\hat{\Sigma} + \mathcal{V})A^{-T}) \\ & \times \phi(y^x; \mu^{\alpha,y}_{m,n+1}(q',y), \mathcal{W} + \Sigma^{\alpha,y}_{m,n+1}(q',y)) \\ \mu^{\alpha}_{y,q',m,i,n+1}(q) = {} & \Sigma^{\alpha}_{y,q',m,i,n+1}(q) \left[\left(\Sigma^I_i(q)\right)^{-1} \mu^I_i(q) \right. \\ & \left. + A^T \left(\hat{\Sigma} + \mathcal{V}\right)^{-1} (\hat{\mu} - g(q,u,q')) \right] \\ \Sigma^{\alpha}_{y,q',m,i,n+1}(q) = {} & \left[\left(\Sigma^I_i(q)\right)^{-1} + A^T \left(\hat{\Sigma} + \mathcal{V}\right)^{-1} A \right]^{-1}, \end{aligned}$$

and

$$\begin{aligned} \hat{\mu} &= \hat{\Sigma} \left[\mathcal{W}^{-1} y^x + \left(\Sigma^{\alpha,y}_{m,n+1}(q')\right)^{-1} \mu^{\alpha,y}_{m,n+1}(q') \right] \\ \hat{\Sigma} &= \left[\mathcal{W}^{-1} + \left(\Sigma^{\alpha,y}_{m,n+1}(q')\right)^{-1} \right]^{-1}. \end{aligned}$$

Proof. The first part of the theorem is by induction. At time N, $\alpha_{N,g}(x,q) = \sum_{i=1}^{I_q} w_i^I(q)\phi(x;\mu_i^I(q),\Sigma_i^I(q))$ by definition, and is therefore a Gaussian mixture. Assuming that $\alpha_{j,g}(x,q)$ is a Gaussian mixture for all $i = N-1,\ldots,n+1$, $\alpha_{n,g}(x,q)$ is

$$\sum_{y\in\mathcal{Y}} \alpha_{y,u,\sigma}^{g}(x,q)$$

which is again a Gaussian mixture by Lemma 5.7, and allowing for an abuse of notation via the summation over $\mathcal{Y}$.

The second part of the theorem follows directly from the derivation provided in the proof of Lemma 5.7. ■

The concern over the validity of a Gaussian representation of the α-functions is addressed by discretizing the space $\mathcal{Y}$, as discussed in Section 5.6. Because the α-functions can only be derived and stored in practice for a finite number of observations, we proposed a discretization scheme over a region $\overline{K} \supset K$. For the Gaussian mixture approximation, this requires a new observation function τ_y^g, with

$$\tau_y^g(w|s,u) = \begin{cases} \tau_y(\Theta(w)|s,u), & \text{if } w \in \mathcal{Y}_\delta \\ 1 - \sum_{\overline{w}\in\mathcal{Y}_\delta} \tau_y(\Theta(\overline{w})|z,u), & \text{if } w = \psi_y \end{cases}, \tag{B.14}$$

and additionally,

$$\tau_y(\Theta(w)|s,u) \approx T_{y^q}(y^q|q,u)\sum_{j=1}^{M_y} c_j\phi_j^y(y_j^{x,i,y^q};x,\mathcal{W}). \tag{B.15}$$

Hence the probability associated with each discretized observation w is approximated by a Gaussian sum.

We can generate the approximation by considering a numerical integration scheme to represent the integral $\tau_y(\Theta(w)|s,u)$. The region $\overline{K}_{i,y^q}$ associated with $\Theta(w) = \overline{K}_{i,y^q} \times y^q$ is subdivided into a finer mesh of points y_j^{x,i,y^q}, $j = 1,\ldots,M_y$. These points are distinct from the representative element y^{x,i,y^q} of $\overline{K}_{i,y^q}$ (with $w = (y^{x,i,y^q}, y^q)$). The weights c_j are chosen according to the numerical integration scheme.

For example, if $\mathcal{Y}^x = \mathbb{R}$, we can divide the interval associated with $\overline{K}_{i,y^q}$ into equally spaced points $y_1^{x,i,y^q}, \ldots, y_{M_y}^{x,i,y^q}$, with spacing Δ_y and y_1^{x,i,y^q} and $y_{M_y}^{x,i,y^q}$ each an endpoint of the interval. Weights c_1 and c_{M_j} are equal to $\frac{\Delta_y}{2}$, and all other c_j equal Δ_y. Then, for $w \in \mathcal{Y}_\delta$,

$$\begin{aligned}\tau_y^g(w|s,u) &= \int_{\Theta(w)} \tau_y(dy|s,u) \\ &\approx T_{y^q}(y^q|q,u) \left[\frac{\Delta_y}{2} \phi(y_1^{x,i,y^q}; x, \mathcal{W}) + \sum_{j=2}^{M_j-1} \Delta_y \phi(y_j^{x,i,y^q}; x, \mathcal{W}) \right. \\ &\qquad \left. + \frac{\Delta_y}{2} \phi(y_{M_y}^{x,i,y^q}; x, \mathcal{W}) \right]. \qquad \text{(B.16)}\end{aligned}$$

We do not consider $w = \psi_y$ when doing actual computations, and so do not need a finite Gaussian sum representation for that case.

References

A

[AAP$^+$07] A. Abate, S. Amin, M. Prandini, J. Lygeros, and S. Sastry. Computational approaches to reachability analysis of stochastic hybrid systems. In *Hybrid Systems: Computation and Control*, pages 4–17. 2007. 28, 81

[ADB11] A. Abate, A. D'Innocenzo, and M. D. Benedetto. Approximate abstractions of stochastic hybrid systems. *IEEE Transactions on Automatic control*, 56(11):2688 – 2694, 2011. 8

[AHS96] R. Alur, T. A. Henzinger, and E. D. Sontag. *Hybrid Systems III: Verification and Control*, volume 3 of *Lecture Notes in Computer Science*. Springer, 1996. 4

[AKLP10] A. Abate, J.-P. Katoen, J. Lygeros, and M. Prandini. Approximate model checking of stochastic hybrid systems. *European Journal of Control*, 16:624–641, 2010. 8, 81, 82, 90

[AM07] P. J. Antsaklis and A. N. Michel. *A Linear Systems Primer*. Birkhauser, 2007. 39

[APLS08] A. Abate, M. Prandini, J. Lygeros, and S. Sastry. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica*, 44(11):2724–2734, 2008. 6, 7, 18, 20, 23, 26, 51, 71, 139

[ASL93] P. Antsaklis, J. Stiver, and M. Lemmon. Hybrid system modeling and autonomous control systems. In *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 366–392. Springer, 1993. 4

B

[BBM98] M. S. Branicky, V. S. Borkar, and S. K. Mitter. A unified framework for hybrid control: Model and optimal control theory. *IEEE Transactions on Automatic Control*, 43(1):31 – 45, 1998. 4

[Ber05] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, 2005. 7, 10, 19, 57

[BKLPR10] E. Brunskill, L. Kaelbling, T. Lozano-Perez, and N. Roy. Planning in partially-observable switching-mode continuous domains. *Annals of Mathematics and Artificial Intelligence*, 58:185–216, 2010. 76, 82

[BL06] M. L. Bujorianu and J. Lygeros. Toward a general theory of stochastic hybrid systems. In H. Blom and J. Lygeros, editors, *Stochastic Hybrid Systems*, volume 337 of *Lecture Notes in Control and Information Science*, pages 3–30. Springer Berlin Heidelberg, 2006. 6

[BL07] M. Bujorianu and J. Lygeros. New insights on stochastic reachability. In *IEEE Conference on Decision and Control*, pages 6172 – 6177, 2007. 6

[BMSS06] J. Bruhn, P. S. Myles, R. Sneyd, and M. M. R. F. Struys. Depth of anaesthesia monitoring: what's available, what's validated, and what's next? *British Journal of Anaesthesia*, 97(1):85–94, 2006. 2

[BMWDW06] A. Brooks, A. Makarenko, S. Williams, and H. Durrant-Whyte. Parametric POMDPs for planning in continuous state spaces. *Robotics and Autonomous Systems*, 54(11):887–897, 2006. 82

[BO09] L. Blackmore and M. Ono. Convex chance constrained predictive control without sampling. In *AIAA Guidance, Navigation and Control Conference*, 2009. 31, 36

[BOBW10] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams. A probabilistic particle-control approximation of chance-constrained stochastic predictive control. *IEEE Transactions on Robotics*, 26(3):502–517, 2010. 31, 34, 35, 36

[BS96] D. P. Bertsekas and S. E. Shreve. *Stochastic Optimal Control: The Discrete-Time Case*. Athena Scientific, 1996. 7, 50, 52, 59, 154, 156

[Buj04] M. Bujorianu. Extended stochastic hybrid systems and their reachability problem. In *Hybrid Systems: Computation and Control*, pages 234–249, 2004. 6

C

[CM11] T. Caldwell and T. Murphy. Switching mode generation and optimal estimation with application to skid-steering. *Automatica*, 47(1):50–64, 2011. 73, 74

D

[DAT13] J. Ding, A. Abate, and C. Tomlin. Optimal control of partially observable discrete time stochastic hybrid systems for safety specifications. In *American Control Conference*, pages 6231–6236, 2013. 51, 70, 71

[DKS$^+$13] J. Ding, M. Kamgarpour, S. Summers, A. Abate, J. Lygeros, and C. Tomlin. A stochastic games framework for verification and control of discrete time stochastic hybrid systems. *Automatica*, 49(9):2665 – 2674, 2013. 5, 7

[DMA09] G. A. Dumont, A. Martinez, and J. M. Ansermino. Robust control of depth of anesthesia. *International Journal of Adaptive Control and Signal Processing*, 23(5):435–454, 2009. 2

E

[Ell93] R. J. Elliot. A general recursive discrete-time filter. *Journal of Applied Probability*, 30(3):575–588, 1993. 66

[ES10] T. Erez and W. Smart. A scalable method for solving high-dimensional continuous POMDPs using local approximation. In *26th Conference on Uncertainty in Artificial Intelligence*, 2010. 82

F

[FF11] B. Fornberg and N. Flyer. The Gibbs phenomenon for radial basis functions. In *Gibbs Phenomenon in Various Representations and Applications*, pages 201–224, 2011. 100

[FGM97] E. Fernandez-Gaucherand and S. I. Marcus. Risk-sensitive optimal control of hidden Markov models: Structural results. *IEEE Transactions on Automatic Control*, 42(10):1418–1422, 1997. 52

[FHH$^+$11] M. Fränzle, E. M. Hahn, H. Hermanns, N. Wolovick, and L. Zhang. Measurability and safety verification for stochastic hybrid systems. In *Hybrid Systems: Computation and Control*, pages 43–52, 2011. 81

G

[GdV14] R. Ghaemi and D. del Vecchio. Control for safety specifications of systems with imperfect information on a partial order. *IEEE Transactions on Automatic Control*, 59(4), 2014. 9, 10

[GG10] C. L. Guernic and A. Girard. Reachability analysis of linear systems using support functions. *Nonlinear Analysis: Hybrid Systems*, 4(2):250 – 262, 2010. IFAC World Congress 2008. 8, 151

[Gir05] A. Girard. Reachability of uncertain linear systems using zonotopes. In *Hybrid Systems:Computation and Control*, pages 291–305, 2005. 8, 151

[Gir12] A. Girard. Controller synthesis for safety and reachability via approximate bisimulation. *Automatica*, 48(5):947–953, 2012. 8

[GLQ06] Y. Gao, J. Lygeros, and M. Quincapoix. The reachability problem for uncertain hybrid systems revisited: a viability theory perspective. In *Hybrid systems: Computation and Control*, pages 242–256, 2006. 4, 5

[GVC12] F. Gavilan, R. Vazquez, and E. F. Camacho. Chance-constrained model predictive control for spacecraft rendezvous with disturbance estimation. *Control Engineering Practice*, 20(2):111 – 122, 2012. 31

H

[HLS00] J. Hu, J. Lygeros, and S. Sastry. Towards a theory of stochastic hybrid systems. In *Hybrid Systems: Computation and Control*, pages 160–173, 2000. 5

[HTRM12] E. N. Hartley, P. A. Trodden, A. G. Richards, and J. M. Maciejowski. Model predictive control system design and implementation for spacecraft rendezvous. *Control Engineering Practice*, 20(7):695 – 713, 2012. 31

[HW04] M. W. Hofbaur and B. C. Williams. Hybrid estimation of complex systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(5):2178–2191, 2004. 9

[IBM99] IBM ILOG. *ILOG CPLEX User's Guide*, 1999. 41

J

[JBE94] M. R. James, J. S. Baras, and R. J. Elliot. Risk-sensitive control and dynamic games for partially observed discrete-time nonlinear systems. *IEEE Transactions on Automatic Control*, 39(4):780–792, 1994. 52, 54, 66

K

[Kay12] S. Kaynama. *Scalable Techniques for the computation of viable and reachable sets*. PhD thesis, University of British Columbia, 2012. 143, 144

[KHL08] H. Kurniawati, D. Hsu, and W. S. Lee. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems*, 2008. 82, 143, 152

[KKNP10] M. Kattenbelt, M. Kwiatkowska, G. Norman, and D. Parker. A game-based abstraction-refinement framework for Markov decision processes. *Formal Methods in System Design*, 36(3):246–280, 2010. 8

[KKZ03] X. Koutsoukos, J. Kurien, and F. Zhao. Estimation of distributed hybrid systems using particle filtering methods. In *Hybrid Systems: Computation and Control*, pages 298–313, 2003. 9

[KNP11] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0, verification of probabilistic real-time systems. In *International Conference on Computer Aided Verification*, pages 585–591, 2011. 8

[KO] S. Kaynama and M. Oishi. A modified Ricatti transformation for decentralized computation of the viability kernel under LTI dynamics. *IEEE Transactions on Automatic Control*, 58(11):2878 – 2892. 8

[KR06] X. Koutsoukos and D. Riley. Computational methods for reachability analysis of stochastic hybrid systems. In *Hybrid Systems: Computation and Control*, pages 377–391, 2006. 7

[KSS$^+$13] N. Kariotoglou, S. Summers, T. Summers, M. Kamgarpour, and J. Lygeros. Approximate dynamic programming for stochastic reachability. In *European Control Conference*, pages 584 – 589, 2013. 8, 76, 81

[KV02] A. Kurzhanski and P. Varaiya. Reachability under uncertainty. In *IEEE Conference on Decision and Control*, volume 2, pages 1982–1987, 2002. 5

[KV07] A. Kurzhanskiy and P. Varaiya. Ellipsoidal techniques for reachability analysis of discrete-time linear systems. *IEEE Transactions on Automatic Control*, 50(1):26–38, 2007. 8, 151

L

[LGM01] C. Lusena, J. Goldsmith, and M. Mundhenk. Nonapproximability results for partially observable Markov decision processes. *Journal of Artificial Intelligence Research*, 14:83–103, 2001. 79, 84, 129

[LH12] W. Liu and I. Hwang. A stochastic approximation based state estimation algorithm for stochastic hybrid systems. In *American Control Conference*, pages 312–317, 2012. 9

[LTS99] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3):349–370, 1999. 4, 6

[Lyg04] J. Lygeros. On reachability and minimum cost optimal control. *Automatica*, 40(6):917–927, 2004. 7

M

[MBT05] I. Mitchell, A. Bayen, and C. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, 2005. 7

[Mit08] I. Mitchell. The flexible, extensible and efficient toolbox of level set methods. *Journal of Scientific Computing*, 35:300–329, 2008. 7

[MKM$^+$13] J. Maidens, S. Kaynama, I. Mitchell, M. Oishi, and G. Dumont. Lagrangian methods for approximating the viability kernel in high-dimensinoal systems. *Automatica*, 49(7):2017 – 2029, 2013. 8, 153

[MSAC02] N. G. M. Sanjeev Arulampalam, Simon Maskell and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2), 2002. 66

[MT02] I. Mitchell and C. J. Tomlin. Level set methods for computation in hybrid systems. In *Hybrid Systems: Computation and Control*, pages 310–323, 2002. 7

[MT05] I. Mitchell and J. Templeton. A toolbox of Hamilton-Jacobi solvers for analysis of nondeterministic continuous and hybrid systems. In *Hybrid Systems: Computation and Control*, pages 480–494. 2005. 7

N

[NAS06] Overview of the DART mishap investigation results. http://www.nasa.gov/pdf/148072main_DART_mishap_overview.pdf, 2006. 2

[NAS07] DART mission overview. http://www.nasa.gov/mission_pages/dart/main/index.html, 2007. xiii, 1, 2

[NJ00] A. Y. Ng and M. Jordan. PEGASUS: a policy search method for large MDPs and POMDPs. In *Conference on Uncertainty in Artificial Intelligence*, pages 406–415, 2000. 82

[NPR98] V. Norkin, G. C. Pflug, and A. Ruszczyński. A branch and bound method for stochastic global optimization. *Mathematical Programming*, 83(1-3):425–450, 1998. 152

[NS06] A. Nemirovski and A. Shapiro. Scenario approximations of chance constraints. In *Probabilistic and Randomized Methods for Design under Uncertainty*, pages 3–47. Springer, 2006. 31

O

[OBW10] M. Ono, L. Blackmore, and B. Williams. Chance constrained finite horizon optimal control with nonconvex constraints. In *American Control Conference*, Baltimore, Maryland, June 2010. 48

[OMAM11] M. M. K. Oishi, N. Matni, A. Ashoori, and M. J. McKeown. Switching restrictions for stability despite switching delay: application to switched tracking tasks in Parkinson's disease. *Journal of Nonlinear Systems and Applications*, pages 16–25, 2011. 4

P

[PDCK11] H. Park, S. Di Cairano, and I. Kolmanovsky. Model predictive control for spacecraft rendezvous and docking with a rotating/tumbling platform and for debris avoidance. In *American Control Conference*, pages 1922–1927, 2011. 31

[PGT06] J. Pineau, G. Gordon, and S. Thrun. Anytime point-based approximations for large POMDPs. *Journal of Artificial Intelligence Research*, 27:335–380, 2006. 57, 82, 85, 108

[PH06] M. Prandini and J. Hu. *Stochastic Reachability: Theoretical Foundations and Numerical Approximation*, pages 107–139. Lecture Notes in Control and Information Sciences. Springer Verlag, 2006. 6

[Pow11] W. B. Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality.* John Wiley and Sons, 2011. 25

[PS91] J. Park and I. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 3:246 – 257, 1991. 100

[PTKLP10] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez. Belief space planning assuming maximum likelihood observations. In *Robotics: Science and Systems*, 2010. 82

[PTKLP11] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez. Efficient planning in non-Gaussian belief spaces and its application to robot grasping. In *15th International Symposium on Robotics Research*, 2011. 82

[Put05] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley and Sons, 2005. 19, 20, 23

[PVSP06] J. M. Porta, N. Vlassis, M. T. Spain, and P. Poupart. Point-based value iteration for continuous POMDPs. *Journal of Machine Learning Research*, 7:2329–2367, 2006. 76, 82, 86, 120, 138

R

[Res02] S. I. Resnick. *Adventures in Stochastic Processes.* Birkhäuser, 2002. 5, 17, 24

[RPPCd08] S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa. Online planning algorithms for POMDPs. *Journal of Artificial Intelligence Research*, 32:663 – 704, 2008. 143, 152

S

[SA13] S. Soudjani and A. Abate. Adaptive and sequential gridding procedures for the abstraction and verification of stochastic processes. *SIAM Journal on Applied Dynamical Systems*, 12(2):921–956, 2013. 8, 81, 126, 134

[SB10] J. Skaf and S. P. Boyd. Design of affine controllers via convex optimization. *IEEE Transactions on Automatic Control*, 55(11):2476–2487, November 2010. 33, 38

[Shi64] A. Shiryaev. On Markov sufficient statistics in nonadditive Bayes problems of sequential analysis. *Theory of Probability and its Applications*, 9(4):604–618, 1964. 50

[SL10] S. Summers and J. Lygeros. Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem. *Automatica*, 46(12):1951–1961, 2010. 7, 23, 26

[Son71] E. Sondik. *The optimal control of partially observable Markov processes*. PhD thesis, Stanford University, 1971. 79, 84

[SPK13] G. Shani, J. Pineau, and R. Kaplow. A survey of point-based POMDP solvers. *Autonomous Agents and Multi-Agent Systems*, 27(1):1–51, 2013. 24, 82, 85, 107

[SS04] T. Smith and R. Simmons. Heuristic search value iteration for POMDPs. In *Conference on Uncertainty in Artificial Intelligence*, 2004. 152

[SS05] E. Stein and R. Shakarchi. *Real Analysis: Measure Theory, Integration, and Hilbert Spaces*. Princeton Lectures in Analysis. Princeton University Press, 2005. 65

[Ste94] R. F. Stengel. *Optimal control and estimation*. Dover, 1994. 7

[SV05] M. Spaan and N. Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artifical Intelligence Research*, pages 195–220, 2005. 82

T

[Thr00] S. Thrun. Monte carlo POMDPs. In *Advances in Neural Information Processing Systems 12*, pages 1064–1070, 2000. 82

[TLS00] C. Tomlin, J. Lygeros, and S. Sastry. A game theoretic approach to controller design for hybrid systems. *Proceedings of the IEEE*, 88(7), 2000. 6

[TMBO03] C. Tomlin, I. Mitchell, A. Bayen, and M. Oishi. Computational techniques for the verification and control of hybrid systems. In *Proceedings of the IEEE*, volume 91, pages 986–1001, 2003. 7

[TMKA13] I. Tkachev, A. Mereacre, J. Katoen, and A. Abate. Quantitative automata-based controller synthesis for non-autonomous stochastic hybrid systems. In *Hybrid Systems: Computation and Control*, pages 293–302, 2013. 23, 52

[TPS98] C. Tomlin, G. J. Pappas, and S. Sastry. Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Transaction on Automatic Control*, 43(4):509 – 521, 1998. 4

V

[VdV12] R. Verma and D. del Vecchio. Safety control of hidden mode hybrid systems. *IEEE Transactions on Automatic Control*, 57(1):62–77, 2012. 9, 10

[VT11] M. P. Vitus and C. J. Tomlin. Closed-loop belief space planning for linear, Gaussian systems. In *IEEE International Conference on Robotics and Automation*, pages 2152–2159, 2011. 31, 36

W

[Wie89] W. Wiesel. *Spaceflight Dynamics*. McGraw-Hill, New York, 1989. 30, 39

[WKBE12] A. Weiss, I. Kolmanovsky, M. Baldwin, and R. S. Erwin. Model predictive control of three dimensional relative motion. In *American Control Conference*, 2012. 31, 39, 41

Z

[ZFM10] E. Zhou, M. Fu, and S. Marcus. Solving continuous-state POMDPs via density projection. *IEEE Transactions on Automatic Control*, 55(5):1101–1116, 2010. 82

[ZK10] K. Zhang and J. Kwok. Simplifying mixture models through function approximation. *IEEE Transactions on Neural Networks*, 21(4):644–658, 2010. 120

[ZTA14] M. Zamani, I. Tkachev, and A. Abate. Bisimilar symbolic models for stochastic control systems without state-space discretization. In *Hybrid Systems: Computation and Control*, pages 41 – 50, 2014. 81