Title
Methods for Reachability-based Hybrid Controller Design

Author
Ding, Jerry

Publication Date
2012

Peer reviewed|Thesis/dissertation

Methods for Reachability-based Hybrid Controller Design

by

Jerry Ding

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Claire Tomlin, Chair
Professor S. Shankar Sastry
Professor Venkat Anantharam
Professor Lawrence C. Evans

Spring 2012

Methods for Reachability-based Hybrid Controller Design

Abstract

Methods for Reachability-based Hybrid Controller Design

by

Jerry Ding

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Claire Tomlin, Chair

With the increasing complexity of systems found in practical applications, the problem of controller design is often approached in a hierarchical fashion, with discrete abstractions and design methods used to satisfy high level task specifications, and continuous abstractions and design techniques used to satisfy low level control objectives. Although such a separation allows the application of mature theoretical and computational tools from the realms of computer science and control theory, the task of ensuring desired closed-loop behaviors, which results from the composition between discrete and continuous designs, often requires costly and time consuming verification and validation. This problem becomes especially acute in safety-critical applications, in which design specifications are often subject to rigorous industry standards and government regulations. Hybrid systems, which feature state trajectories evolving on a combination of discrete and continuous state spaces, have been proposed as a possible approach to reconcile the analysis and design techniques from the discrete and continuous domains under a rigorous theoretical framework. However, designing controllers for general classes of hybrid systems is a highly nontrivial task, as such a design problem inherits both the difficulty of nonlinear control, as well as the range of theoretical and computational issues introduced by the consideration of discrete switching.

This dissertation describes several efforts aimed towards the development of theoretical analysis tools and computational synthesis techniques to facilitate the systematic design of feedback control policies satisfying safety and target attainability specifications with respect to subclasses of hybrid system models. The main types of problems we consider are safety/invariance problems, which involve keeping the closed-loop state trajectory within a safe set in the hybrid state space, and reach-avoid problems, which involve driving the state trajectory into a target set subject to a safety constraint. These problems are addressed within the context of continuous time switched nonlinear systems and discrete time stochastic hybrid systems, as motivated by application scenarios arising in autonomous vehicle control and air traffic management.

First, we provide several design techniques and synthesis algorithms for deterministic reachability problems formulated in the setting of switched nonlinear systems, with controlled switches between discrete modes, and bounded continuous disturbances. For scenarios in which the mode transitions proceed in a known sequence, a method is discussed for designing controllers to satisfy

sequential reachability specifications, consisting of a temporally ordered sequence of invariance and reach-avoid objectives. In particular, we use continuous time reachable sets to inform choices of feedback control policies within each discrete mode to satisfy both individual reachability objectives and compatibility conditions between successive modes. This technique is illustrated through an example of maneuver sequence design for automated aerial refueling of unmanned aerial vehicles. For scenarios in which the modes of a switched system can be freely selected, we describe an approach for the automated synthesis of feedback control policies achieving safety and reach-avoid objectives, under a sampled data setting. This synthesis technique proceeds by a structured reachability computation which retains information about the choice of switching controls at each discrete time instant, resulting in a set-valued policy represented in terms of a finite collection of reachable sets. Experimental results from the implementation of such control policies on a quadrotor platform to track a moving ground target show strong robustness properties in the presence of significant disturbances.

Second, we provide theoretical and computational results on stochastic game and partial information formulations of probabilistic reachability problems. In the setting of a discrete time stochastic hybrid game model, zero-sum dynamic game formulations of probabilistic safety and reach-avoid problems are considered. Under an asymmetric information pattern favoring the adversary, we prove dynamic programming results for the computation of finite horizon max-min safety and reach-avoid probabilities and synthesis of deterministic max-min control policies. The implications of alternative information patterns and infinite horizon formulations are also discussed. In particular, it is shown that under a symmetric information pattern, equilibrium solutions are in general found within the class of randomized policies. The utility of this approach is illustrated through an example of pairwise aircraft conflict resolution, with a probabilistic model of wind effects. In the setting of a partially observable discrete time stochastic hybrid system, we provide a characterization of the optimal solution to partial information probabilistic safety and reach-avoid problems, which have nonstandard multiplicative and sum-multiplicative cost structures. In particular, these problems are shown to be equivalent to terminal cost and additive cost problems, by augmenting the hybrid state space with a binary random variable capturing the safety of past state evolution. Using this result, we derive a sufficient statistic in terms of a set of Bayesian filtering equations, along with an abstract dynamic programming algorithm for computing the optimal safety and reach-avoid probabilities. The practical implementation of the estimation and control algorithms, however, will depend on the existence of finite dimensional representations or approximations of the hybrid probability distribution.

To my parents, friends, and extended family

Acknowledgments

The past six years of graduate studies at Berkeley has been a long journey for me, one which I was not always sure I would complete. What has sustained me throughout this memorable part of my life is the guidance, support, encouragement, and friendship of the many people that I have had the privilege of meeting along the way. To them, I owe a deep debt of gratitude.

The research described in this dissertation had its beginnings with Professor Shankar Sastry who took me under his wings in the Spring of 2007 on the automated aerial refueling project. Looking back, I was an otherwise unremarkable first year student aside from some good grades during undergraduate studies, but little of the research background or formal mathematical preparations of many of my peers. I am very much grateful that he was able to provide me with this entryway into the world of research, and also to introduce me to the concept of reachability, which has fascinated me ever since. If Professor Sastry was the one who opened the door for me, then Dr. Jonathan Sprinkle, my mentor on the project and now a Professor at University of Arizona, was the one who helped me walk through the door. It is with fond memory that I recall his unrelenting sense of optimism and healthy dose of humor, which made the otherwise daunting task of learning to do research while doing it much more pleasant. His patience and tirelessness in answering my numerous questions and his invaluable help in writing my first conference paper was in large part the reason for my first fruits of research.

After these initial steps, I count it as one of my great fortunes to have been able to join the research group of my current advisor, Professor Claire Tomlin. It is under her patient guidance and encouragement that I gradually developed maturity in my thought process and understanding of the art of academic research. There have been many intervals between projects during which I spent a significant amount of time brainstorming for ideas and exploring different avenues, not all of which turned out to be viable in the end. I very much appreciate her understanding and support through all these periods, and also for her encouragements when things just simply did not turn out the way I hoped for. My parents have always reminded me that I am not the quickest of learners and that it really took a special kind of patience to teach me what to other children might have been mundane tasks. For this reason alone, I would like to say these words from the bottom of my heart, "Thank you, Claire, for your special kind of patience."

I would also like to express my profound appreciation to Professors Craig Evans and Venkat Anantharam, who have been very gracious in agreeing to be part of my dissertation committee, and in taking time from their busy schedules to offer their valuable comments and suggestions during my qualifying examination. It is Professor Evans's lecture notes on optimal control and dynamic games which constituted the foundations of my knowledge about these respective topics. His book on partial differential equations has also been indispensible in my work related to reachability calculations using Hamilton-Jacobi equations in the first part of this dissertation. My interest in stochastic optimal control, which forms the core of the discussion in the second part of this dissertation, is in large part sparked by the lectures of Professor Anantharam on the same topic. I have tremendous respect for his commanding knowledge of this area of research and also the extraordinary care with which he imparted them to us during class.

The work to be discussed in the subsequent chapters is the result of collaboration with, as well as support and guidance provided by many coauthors, colleagues, and mentors. On the aerial refueling project, as presented in chapter 2, I give my sincere thanks to our collaborators at Boeing Research & Technology, including Jim Barhorst, Jim Paunicka, and Doug Stuart, for their important role in formulating the automated aerial refueling scenario used in our research study, and also to members of the Air Force Research Laboratory (AFRL) for their valuable comments and feedback over the course of the project. Many of the research ideas grounding our effort arose from discussions at David Homan's yearly meetings on Verification and Validation at Wright-Patterson Air Force Base in Dayton, OH.

In the process of learning about Hamilton-Jacobi reachability for the work on automated aerial refueling, it is my privilege to have been under the guidance of Professor Alex Bayen of the Civil Engineering department here at Berkeley. I am grateful for his kindness and patience in explaining to me the basics of reachability calculations, as well as his generosity in offering experiences and insights from his own work on the use of reachable sets for safety analysis in flight control systems. I would also like to thank Professor Ian Mitchell of University of British Columbia for his wonderful help on many technical issues related to the computation of reachable sets using level set methods, as well as our fascinating discussions during his visit to Berkeley in the Spring of 2011. The reachability computations described in the first part of this dissertation would not have been possible without his Toolbox of Level Set Methods.

The implementation of the controller synthesis algorithms on the STARMAC platform, as described in chapter 3, is in large part thanks to the hard work and dedication of Eugene Li, who spent many countless hours in the lab writing and debugging the code, and running flight tests. This was in addition to his classes and regular orchestra practices. The fact that we were able to go from theory on paper to implementation and experimental results in little more than two months speaks to his extraordinary efforts. On the same note, both Eugene and I would like to express our gratitude for all the help and support we received from the STARMAC group, including Patrick Bouffard, Jeremy Gillula, Haomiao Huang, Tony Mercer, and Michael Vitus.

For the work on stochastic hybrid games and partially observable stochastic hybrid systems, as discussed in chapter 4 and 5, the initial ideas came from a flurry of correspondences with Professor Alessandro Abate of Delft University of Technology starting in April of 2010. The growth and development of these ideas have much to do with the detailed comments and follow-ups by Alessandro as our exchanges continued through the summer, and then the subsequent two years. It was also thanks to Alessandro that we bridged the work on stochastic games with the parallel effort of Maryam Kamgarpour and Sean Summers in January of 2011. I would like to thank Maryam and Sean for our enlightening discussions on the various theoretical issues related to stochastic game formulations of hybrid reachability problems and our sometimes spirited debates. They provided much of the impetus as I derived the set of theoretical results which now appear, after several revisions, in their current form in Chapter 4. The revisions also include many of the insightful comments by Professor John Lygeros of ETH Zürich. In particular, the inclusion of the discussion on Nash formulations of the probabilistic reach-avoid problem were in large part motivated by what Professor Lygeros correctedly pointed out as a glaring omission on our part in the initial set of results.

Over the course of these research projects, I have been generously supported by a number of funding sources. The work on automated aerial refueling as described in chapter 2 was supported in part by the "Certification Technologies for Flight Critical Systems (CerTA FCS)" project, Air Force Research Labs (AFRL), through a contract with Boeing Research & Technology; and in part by the Center for Hybrid and Embedded Software Systems (CHESS) at UC Berkeley, which receives support from the National Science Foundation (NSF awards #CCR-0225610 (ITR),#0720882 (CSR-EHS: PRET), #0647591 (CSR-SGER), and #0720841 (CSR-CPS)), the U.S. Army Research Office (ARO #W911NF-07-2-0019), U.S. Air Force Office of Scientific Research (AFOSR) awards MURI #FA9550-06-0312 and AF-TRUST #FA9550-06-1-0244, AFRL, the State of California Micro Program, and the following companies: Agilent, Bosch, DGIST, Lockheed Martin, National Instruments, and Toyota. Additional support was provided by AFOSR Award #FA9550-091-0519 titled "Modeling of Embedded Human Systems," and NSF awards CNS-0915010 and CNS-0930919. The work on controller synthesis for switched systems as described in chapter 3 and the work on stochastic game formulations of probabilistic reachability calculations as described in chapter 4 were supported by the MURI - Frameworks and Tools for High Confidence Design of Adaptive, Distributed Embedded Control Systems project administered by the Air Force Office of Scientific Research (AFOSR) under Grant #FA9550-06-1-0312. The work on partially observable stochastic hybrid systems as described in chapter 5 has been supported by NSF Large CPS ActionWebs Project under Grant #CNS-0931843.

Beyond the scope of this dissertation, I have also had the great pleasure to work with several outstanding collaborators and mentors on a number of other research efforts. I would like to thank Gabe Hoffmann of Palo Alto Research Center, Professor Steven Waslander of the University of Waterloo, Professor Carlo Fischione of KTH, and Professor Alberto Sangiovanni-Vincentelli of the EECS department here at Berkeley for their valuable insights, guidance, and support on a project studying the effect of communication delays on the performance of networked UAVs. In a separate research direction, I would like to thank Haomiao Huang, Selina Pan, Ryo Takei, Professor Ian Mitchell of the University of British Columbia, Professor Dušan Stipanović of the University of Illinois at Urbana-Champaign, and Professor Wei Zhang of Ohio State University for the wonderful working experiences and personal interactions during our joint efforts on several projects related to pursuit-evasion differential games with state constraints. These projects have all in their own unique way contributed to the broadening of my knowledge of different fields, as well as my development as an academic researcher.

I am very much grateful to members of the hybrid systems lab, both past and present, for providing me with a relaxed and yet highly productive working atmosphere. First, I would like to thank Gabe Hoffmann and Steven Waslander for being gracious hosts when I visited Claire's lab at Stanford in the Fall of 2007. Their friendly and easygoing manners made me feel comfortable and at home during those early days as a new member of the research group. As Claire moved more permanently to Berkeley, members of our group have included, at various points in time, Alessandro Abate, Anil Aswani, Maximilian Balandat, Patrick Bouffard, Young-Hwan Chang, Mo Chen, Vera Dadok, Roel Dobbe, Jeremy Gillula, Harendra Guturu, Souleiman Itani, Qie Hu, Haomiao Huang, Maryam Kamgarpour, Eugene Li, Neal Master, Tony Mercer, Selina Pan, Pangun Park, Ryo Takei, Michael Vitus, Insoon Yang, and Wei Zhang. I would like to thank them for

all the interesting discussions and research presentations during our weekly group meeting over the years, which have added both depth and breadth to my knowledge, as well as the numerous constructive comments, feedback, and suggestions which have had an important influcence in the way I formulate new research directions and also in how I approach ongoing projects. The many social events, both inside and outside the lab, and the experiences of going on conference trips with members of the group, have enriched both my academic and personal life and have left a mosaic of pleasant memories.

Aside from research, I unexpectedly discovered, during my time at Berkeley, one of the great joys of my life – teaching. This came about during the two semesters when I worked as a teaching assistant for EE 20N, in the Fall of 2006 and the Spring of 2007. Dr. Babak Ayazifar, who was the instructor for the class, and who really much prefers to be called "Babak" by his students and his teaching assistants, is a source of inspiration for me. It was from observing Babak's natural and engaging style of teaching that I gradually broke through my psychological fears of speaking in public, a barrier which has been part of my defining characteristic from a very early age. Furthermore, his emphasis on intuition, first principles, and incremental build-up has stayed with me throughout the past six years, not only in how I present technical material, but also in the way I conduct research on a new topic. During that time, I also had the honor to work along side some very capable fellow teaching assistants, including Eric Battenberg, Alvise Bonivento, Christophe Choumert, Siddarth Jain, Mary Knox, Howard Lei, Marghoob Muhiyuddin, Gap Thirathon, and Yang Zhao. I am grateful for their insights, experience, and guidance both during the difficult first semester of getting adjusted to my role as a teaching assistant, and also the second semester of serving as a head TA. A special thanks goes to Sid, Mary, Howard, and Gap for their tremendous help during the hectic Spring 2007 semester, when we operated under severe shortage in staff and a structural overhaul in the lab material. I personally view the department teaching award that semester as a recognition for the entire teaching staff, not any particular individual. Last but not least, the students of EE 20N have been an absolute pleasure to work with, not only for their intellectual curiosity and their willingness to engage on a personal level with the teaching staff, but also for their understanding during the inevitable bumpy spots. I could only hope that many of them have gone on to bigger and brighter futures.

What has allowed me to concentrate on my studies, teaching, and research has been the tireless efforts of the administrative staff, including Ruth Gjerde, Therese George, Gary Given, Jessica Gamble, and Shirley Salanio. To Ruth, thank you for being simply one of the nicest persons I have ever met. Just to remember how you patiently answered my questions in your kind and even-tempered manner brings comfort to my heart. I sincerely hope that you have found peace and happiness in your retirement. To Therese, you were our pillar of support in EE 20N. Without you, I could not imagine how things might have turned out with all of Babak's administrative disorganizations. To Gary and Jessica, thank you for shielding us from the numerous bureaucratic complications which happen behind the scenes in our lives as research assistants. To Shirley, I am very grateful for all your timely reminders and understanding as I made hasty arrangements for the completion of my degree requirements. Finally, to the many in the administrative staff whom I have not had significant opportunities to interact with, I offer these paraphrased words of my favorite singer and songwriter: why does the moon shine so bright, it is because the sun

illuminates the moon from afar. For illuminating us, thank you so very much.

As I am getting closer towards the end of this journey, I have become increasingly reflective of what has made this journey possible, and as such, I would like to give some special acknowledgements to several people who were there at the very beginning. First, I would like to thank my parents for encouraging me to apply for graduate studies at Berkeley when I really did not think I would have a chance. Their love and support throughout the past six years is a wonderful personal story that I will briefly describe somewhat later on. Second, I would like to thank Professor Ross Barmish of the University of Wisconsin at Madison for his kind words of advice to pursue my studies at Berkeley. The brief but sincere meeting with Professor Barmish during visit day at Madison in the Spring of 2006 provided me with the much needed push to make the leap of faith, and his parting gift of contacting Professor Sastry on my behalf was much more than what I could have asked for.

The last part of these acknowledgements are somewhat more personal. First, I would like to thank Anil Aswani for his close friendship over the past several years. Perhaps due to my naturally inward personality, there have been only a few individuals over the course of my life whom I could open up to completely without any fears and reservations, and Anil is one of them. In the past year, it has become somewhat of a tradition for us on Mondays and Wednesdays to grab dinner some place (most often at Celia's), as an excuse to discuss research, outside interests, personal life, or just any topic which the conversation happen to stray into. These conversations have provided both of us with some much needed time to freely express our inner worries, doubts, and anxieties, as well as to share our successes and happiness. Second, I would like to give my sincere thanks to Kathleen Simpson, my landlord of six years, and my adopted grandmother away from home. During that very first summer when I arrived in Berkeley, I fell in love with her little place up in the hills and just never left. She has been every bit as kind and caring for me over the years as though I were part of her own family.

In closing, I would like to thank my parents Kecheng Ding and Weimin Xia, and my extended family for all the heart and soul they have poured into me in the long process of raising me to become the person I am today. As our little family of three moved from one place to another through three continents over the course of a decade, home for me is no longer a physical place but rather an abstract concept – that home is wherever my parents are. In moments of doubt, they have been there to soothe away my fears; in moments of failure, they have been there to put things in perspective; in moments when I wanted to turn back, they have been there to tell me how far I have come and to encourage me to give my best effort. Most of all, I would like to thank my parents for believing in me when I often did not believe in myself. I also would like to express my heartfelt gratitude to my extended family back in Shanghai, especially my aunts Xia Wei Jie and Li Wen Lan, who raised me as though their own child, and my cousin Xia Yi Jia, who grew up with me as though my own sister in the winding back alleyways. I have many fond memories of them that I will treasure for a lifetime.

Jerry Ding
Berkeley, California
May, 2012

Contents

List of Figures

List of Abbreviations, Notations, and Symbols

Abbreviations

AAR	Automated Aerial Refueling
DTSHS	Discrete Time Stochastic Hybrid System
DTSHG	Discrete Time Stochastic Hybrid Game
H-J	Hamilton-Jacobi
HJB	Hamilton-Jacobi-Bellman
HJI	Hamilton-Jacobi-Isaacs
IMM	Interacting Multiple Model
JLS	Jump Linear System
LEG	Linear Exponential Gaussian
LQG	Linear Quadratic Gaussian
MMSE	Minimum Mean Square Error
MSI	Minimum Separation Infringement
PDE	Partial Differential Equation
POdtSHS	Partially Observable Discrete Time Stochastic Hybrid System
POMDP	Partially Observable Markov Decision Processes
STARMAC	Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control
UAV	Unmanned Aerial Vehicle

Common Notations

$\mathbb{R}^n$	n-dimensional Euclidean space
$\mathbb{N}$	Space of natural numbers
$\mathscr{B}(S)$	Borel σ-algebra of topological space S
$\mathscr{P}(S)$	Space of probability measures over S
2^S	Power set of S
$\emptyset$	Empty set
$B(x,r)$	Ball of radius r centered on x
sgn	Signum function
J	Cost function of optimal control problem
S^C	Complement of set S
v^T, M^T	Transpose of vector v or matrix M
$f^N, \mathscr{T}^N$	N-times composition of a function f or operator $\mathscr{T}$

$\mathcal{N}(\bar{x},\Sigma)$	Normal distribution with mean $\bar{x}$ and covariance Σ
$\mathcal{U}[c_1,c_2]$	Uniform distribution over interval of real line $[c_1,c_2]$

Hybrid System Modeling

$\mathscr{H}$	Hybrid system
q,Q	Discrete state, discrete state space
x,X	Continuous state, continuous state space
s,S	Hybrid state, hybrid state space
o,O	Discrete observation, discrete observation space
y,Y	Continuous observation, continuous observation space
z,Z	Observation, observation space
σ,Σ_1	Discrete control input, discrete control space
$\hat{\sigma},\Sigma_2$	Discrete disturbance input, discrete disturbance space
Σ	Discrete input space $\Sigma_1 \times \Sigma_2$
u,U	Continuous control input, continuous control space
$\tilde{u},\tilde{U}$	Quantized control input, quantized control space
d,D	Continuous disturbance input, continuous disturbance space
In	Continuous input space $U \times D$
a,C_a	Player I input, player I input space
b,C_b	Player II input, player II input space
$Init$	Set of permissible initial conditions
f	Vector field, righthand side of difference equation
Dom	Domain of discrete mode
$Reset$	Reset relation
δ	Discrete transition relation
ν_x	Continuous state transition kernel
ν_q	Discrete state transition kernel
ν_r	Reset transition kernel
ν	Hybrid state transition kernel
ζ	Observation kernel
p_0	Initial state probability distribution
p_q	Discrete state transition probability mass function
p_o	Discrete observation probability mass function
p_x	Continuous state transition probability density function
p_r	Reset transition probability density function
p_z	Hybrid observation probability density function
p_s	Hybrid state transition probability density function
w	Process noise random variable
t	Continuous time variable, taking values in set of non-negative reals
T	Continuous time horizon, sampling interval
Δt	Discretization step
k	Discrete time variable, taking values in set of non-negative integers
N	Discrete time horizon
τ	Hybrid time trajectory

L_k	Interval in hybrid time trajectory
$\mathscr{D}_T$	Set of measurable disturbance realizations over time interval $[0,T]$
i_k, I_k	Information vector, information space at time k
Ω	Sample space of hybrid state trajectories $S \times S \times \cdots$
Ω_k	Sample space of hybrid state trajectories and information vectors $S^{k+1} \times I_k$

Control Policies

F	Switching input feedback policy
K	Continuous input feedback policy
$\mu, \mathscr{M}$	Markov policy depending only on hybrid state at each discrete time instant, space of Markov policies
γ, Γ	Markov strategy depending on hybrid state and control or disturbance input at each discrete time instant, space of Markov strategies
π', Π'	Randomized non-Markov observation-based policies, space of such policies
π, Π	Deterministic non-Markov observation-based policies, space of such policies
$\hat{\pi}', \hat{\Pi}'$	Randomized non-Markov policies for equivalent perfect state information model, space of such policies
$\hat{\pi}, \hat{\Pi}$	Deterministic Markov policies for equivalent perfect state information model, space of such policies

Set Specifications

R	Target set
A	Avoid set
W	Safe set, operating constraints

Deterministic Reachability – Switched Systems

$\mathscr{R}$	Capture set, states reachable to target set
$\mathscr{R}_T^{q_i,\tilde{u}}$	Capture set under fixed mode and quantized input over $[0,T]$
$\mathscr{A}$	Unsafe set, states reachable to avoid set
$\mathscr{A}_T^{q_i,\tilde{u}}$	Unsafe set under fixed mode and quantized input over $[0,T]$
$\mathscr{A}_T^H$	Unsafe set for sampled-data switched system over $[0,T]$
$\mathscr{R}\mathscr{A}_T^{q_i,\tilde{u}}$	Reach-avoid set under fixed mode and quantized input over $[0,T]$
$\mathscr{R}\mathscr{A}_T^H$	Reach-avoid set for sampled-data switched system over $[0,T]$
Inv	Invariant subset of safe set
G_{Safe}^N	Horizon-N safe set
G_{Safe}^∞	Infinite horizon safe set
V_k^H	Horizon-k safe set in reachability algorithm for switched systems
G_{RA}^N	Horizon-N reach-avoid set
S_k^H	Horizon-k reach-avoid set in reachability algorithm for switched systems
H	Hamiltonian for deterministic reachability, integral operator for probabilistic reachability
ϕ	Level set representation of reachable sets
$Reach(q)$	Discrete reachability operator, states in Q reachable from q in one step

Probabilistic Reachability – DTSHG

$P_s^{\mu,\gamma}$	Probability measure over sample space Ω, induced by initial condition s, player I policy μ, and player II strategy γ
$E_{s_0}^{\mu,\gamma}$	Expectation with respect to probability measure $P_s^{\mu,\gamma}$
$p_{s_0}^{\mu,\gamma}$	Safety probability given initial condition s_0, player I policy μ, and player II strategy γ
$p_{s_0}^{\mu}$	Worst-case safety probability given initial condition s_0 and player I policy μ
$p_{s_0}^{*}$	Max-min safety probability given initial condition s_0
$r_{s_0}^{\mu,\gamma}$	Reach-avoid probability given initial condition s_0, player I policy μ, and player II strategy γ
$r_{s_0}^{\mu}$	Worst-case reach-avoid probability given initial condition s_0 and player I policy μ
$r_{s_0}^{*}$	Max-min reach-avoid probability given initial condition s_0
$r_{s_0}^{l}$	Lower value of symmetric information reach-avoid problem given initial condition s_0
$r_{s_0}^{u}$	Upper value of symmetric information reach-avoid problem given initial condition s_0
$\mathscr{T}$	Dynamic programming operator for probabilistic reachability
$\mathscr{T}_{\mu,\gamma}$	Recursion operator with respect to fixed player I policy μ and player II strategy γ
$\mathscr{F}$	Set of Borel-measurable functions from S to $[0,1]$
J_k^{*}	Optimal cost-to-go functions in probabilistic reachability algorithms
$J_k^{\mu,\gamma}$	Cost-to-go functions in reachability algorithm for fixed player I policy μ and player II strategy γ
$r_{s_0}^{N}$	Finite horizon max-min reach-avoid probability over $[0,N]$, given initial condition s_0
V_N	Value function representation of $r_{s_0}^{N}$
V_∞	Limit of V_N as $N \to \infty$
$r_{s_0}^{\infty}$	Infinite horizon max-min reach-avoid probability given initial condition s_0
V^*	Value function representation of $r_{s_0}^{\infty}$

Probabilistic Reachability – POdtSHS

$P_k(\pi', p_0)$	Probability measure over sample space Ω_k, induced by initial distribution p_0 and observation-based policy π'
$E_{p_0}^{\pi'}$	Expectation with respect to probability measure $P_N(\pi', p_0)$
$p^{\pi'}(p_0; W)$	Safety probability given initial distribution p_0 and observation-based policy π'
$p^{*}(p_0; W)$	Optimal safety probability given initial distribution p_0
$r^{\pi'}(p_0; R, W')$	Reach-avoid probability given initial distribution p_0 and observation-based policy π'
$r^{*}(p_0; R, W')$	Optimal reach-avoid probability given initial distribution p_0
h_k	History state
$p_{k\|k}$	Conditional distribution of hybrid state
η_k	Sufficient statistic for augmented system
$\tilde{p}_k$	Augmented information state
$\tilde{S}$	Augmented hybrid state space

$\hat{S}$	Augmented information state space
ξ	Map from initial distribution space to augmented initial distribution space
φ	Map from initial distribution space to augmented information state distribution space
Ψ	Information state prediction operator
Φ	Information state update operator

Chapter 1

Introduction

1.1 The Dichotomy Between Discrete and Continuous Abstractions

The task of controller design for modern control systems such as found in aircraft, automobiles, and industrial machinery is a highly complex undertaking. This complexity results in part from the large number of interacting system components, and in part from the wide range of design specifications (e.g. comfort, safety, stability, efficiency) that must be satisfied, often with the expectation of a high degree of reliability. A common approach to controller design for such systems is a layered control architecture in which successively coarser abstractions are employed as one progresses from low level control objectives to high level specifications. While low level control design is often performed using continuous state models (e.g. differential or difference equations) and implemented using analog devices, high level control design is often performed using finite state models (e.g. finite state machines) and implemented using embedded software and electronic devices. In the case of the former, one can take advantage of the rich set of design and analysis methods that has been developed in the realm of control theory, while in the case of the latter, one can take advantage of the numerous efficient algorithms that have been proposed in the realm of computer science. However, there is unfortunately a sparsity of formal design tools at the interface between the two domains. In safety-critical control applications, this presents somewhat of a dilemma, as safety specifications are often described in terms of the closed-loop behavior of the overall system, and hence span the different layers of the control architecture. In particular, the satisfaction of such specifications, which are often determined by rigorous industry standards and government regulation, depends intimately on the interaction between the discrete and continuous layers of control.

To be more concrete, consider the example of conflict detection and resolution in air traffic management. Under current Federal Aviation Administration (FAA) regulations, each aircraft is required to maintain a minimum horizontal and vertical separation distance from other aircraft in the airspace. The problem of conflict detection is one of predicting whether a loss of separation will occur and the problem of conflict resolution is one of executing evasive maneuvers in the event

that a potential conflict is detected. These maneuvers are commonly composed of a discrete set of basic aircraft motions, for example, accelerate, turn, descend, and ascend.

It can be observed that the high level decisions of *when* to initiate conflict resolution maneuvers and *how* the maneuvers should be carried out are both intimately related to the continuous behavior of the aircraft involved in the conflict scenario, in particular the kinematics of each aircraft. At the same time, the conflict resolution problem is not a purely continuous control problem, as the execution of conflict resolution maneuvers depends to a large degree on the design of high level decision protocol. In particular, if the maneuvers are to be carried out according to a pre-defined sequence, then the conflict resolution problem is one of deciding when the aircraft should switch from one motion to the next. On the other hand, if the aircraft is allowed to select freely from a library of basic motions, the problem then becomes one of deciding both the sequence of motions, as well as the times at which to perform the switch. If one were to consider in addition the various uncertainties during the execution of conflict resolution maneuvers, for example the unknown intention of the other aircraft or disturbances to aircraft motion due to wind effects, it is then no longer sufficient to consider open-loop choices of maneuver sequences and switching times. In this case, the conflict resolution problem becomes one of designing a feedback policy for discrete maneuver selection, such that the continuous closed-loop trajectory of the aircraft maintains minimum separation distance at all times.

1.2 High-Confidence Controller Design as Hybrid Reachability Problem

The main focus of this dissertation is on the development of theoretical tools and computational techniques for the design of feedback control policies at the interface between the discrete and continuous layers of the control architecture, with the objective of satisfying certain functional specifications on the closed-loop system behavior. In particular, we will be interested in functional specifications of the following types: 1) *safety*: keep the system state within a prescribed safe set in the system state space over finite or infinite time horizon; 2) *reach-avoid*: drive the system state into a prescribed target set in the system state space within finite time, subject to a constraint that the state trajectory avoids an unsafe set. These specifications are often referred to in the literature collectively as *reachability* specifications. Given that the controller design must be conscious of the discrete nature of high level decision making, as well as the continuous nature of the physical system, a natural modeling framework is that of a hybrid system.

A hybrid system is a dynamical system whose dynamics evolves on a product of discrete and continuous state spaces. The study of such systems within a formal mathematical framework can be traced to the seminal work of Witsenhausen (1966). By now, there is a well-developed body of literature devoted to the modeling and analysis of hybrid systems (see for example Gollu and Varaiya, 1989; Brockett, 1993; Alur et al., 1993; Antsaklis et al., 1993; Nerode and Kohn, 1993; Caines and Wei, 1998; Branicky et al., 1998; Hu et al., 2000). These models have been employed in the study of application scenarios ranging from air traffic management (Sastry et al., 1995; Tomlin

et al., 2002), automotive control (Balluchi et al., 2000), systems biology (Ghosh and Tomlin, 2004; Lincoln and Tiwari, 2004), to unmanned aerial vehicles (Frazzoli et al., 2000; Koo et al., 2001). In certain cases, they are used to capture the interaction between discrete and analog components of the physical system, for example the use of switching elements in the control of electrical and mechanical systems (Aimer et al., 2007). In other cases, they are used to capture sharp changes in the continuous behavior of a dynamical system, for example the operation modes of an automotive engine (Balluchi et al., 2000), or the phases of bipedal walking (Ames et al., 2009). Finally, perhaps most relevant for our discussions, they have been used as a mathematical formalism to integrate discrete and continuous abstractions in a hierarchical control architecture (Gollu and Varaiya, 1989; Lygeros, 1996; Caines and Wei, 1998; Alur et al., 2001).

For the purposes of controller design, a hybrid system model provides us with an abstraction of the interactions between the high level and low level control layers. In particular, the mechanisms for high level decision making can be abstracted in terms of a discrete transition system, while continuous behaviors resulting from high level commands can be abstracted in terms of continuous state models associated with each of the discrete modes. The interactions between the control layers is then captured through the possible dependence of discrete transitions on continuous state variables, as well as the possible dependence of continuous dynamics on discrete state variables. The various sources of uncertainty, such as unmodelled system dynamics or environment disturbances, can be included as exogenous inputs or stochastic noise affecting the discrete or continuous state evolution.

Within the context of a hybrid system model, the problem of designing control policies to satisfy safety or reach-avoid control objectives can be elegantly posed as a reachability problem on the hybrid state space, through proper interpretations of the specifications as constraints on the discrete and continuous states. In the event that the relevant behaviors of the underlying control system are accurately described in terms of the hybrid system model, it can be then expected with a high degree of confidence that the solution to a hybrid reachability problem will satisfy the desired control objectives on the actual system. However, given the inevitable deviations between the complex behaviors of the actual system and the mathematical properties of an abstraction that is tractable for analysis and control, one should not expect that this will completely eliminate the need for formal verification and validation. Instead, what can be hoped for is that through a principled controller design approach based upon mathematical models rather than heuristic insights, one can reduce the prohibitive amount of time and effort that are currently expended on the verification and validation of safety-critical control systems.

1.3 Computational Solutions to Reachability Problems

From the point of view of control theory, it has been recognized that hybrid reachability problems are equivalent to optimal control or dynamic game problems, with real-valued cost functions defined on the hybrid state space (see for example Asarin et al., 1995; Lygeros et al., 1999*b*; Tomlin et al., 2000; Koutsoukos and Riley, 2006; Amin et al., 2006; Mohajerin Esfahani et al., 2011). In fact, this equivalence is not particular to hybrid systems, but is rather a fundamental characteristic

of reachability problems. To illustrate this point, consider a control system of the form $\dot{x} = f(x,u)$, $x(0) = x_0 \in \mathbb{R}^n$. With appropriate assumptions on the vector field f, the solution trajectory $x(\cdot)$ of this system is uniquely determined by the initial condition x_0, and the choice of controls $u(\cdot)$. Given a safe set $W \subset \mathbb{R}^n$, we can use the procedures described in Lygeros et al. (1999*b*) to define a cost function $J(x_0,u)$, such that $J = 1$ if the trajectory corresponding to (x_0,u) satisfies $x(t) \in W$ for every t over the time horizon of interest, and $J = 0$ otherwise. In other words, $J = 1$ if and only if the safety specification is satisfied. Now consider the optimal control problem $J^*(x_0) = \max_u J(x_0,u)$. Then verifying the safety property consists of computing the value J^* and finding the set of initial conditions such that $J^* = 1$, while controller design consists of finding a maximizer for each of these initial conditions.

The advantage of this viewpoint is that finding computational algorithms solving hybrid reachability problems becomes equivalent to finding computational algorithms solving optimal control problems (in the case of a single control agent) and dynamic game problems (in the case of a control and a disturbance). This allows a control engineer to tap into the wealth of knowledge and insight that has accumulated in the respective fields of optimal control and dynamic games. It then comes as little surprise that a significant number of computational algorithms that have been proposed for deterministic or probabilistic reachability, especially in the context of systems with continuous states, are based upon either the dynamic programming principle or the maximum principle (see for example Kurzhanski and Varaiya, 2000; Crück and Saint-Pierre, 2004; Mitchell et al., 2005; Hwang et al., 2005; Koutsoukos and Riley, 2006; Abate et al., 2007). In particular, the computational algorithms for controller synthesis discussed in this dissertation are primarily based upon the dynamic programming principle.

Despite the mathematical elegance of an optimal control or dynamic game formulation of the hybrid reachability problem, solving such a problem for general hybrid systems is a significant challenge (Branicky et al., 1998). In particular, hybrid optimal control problems feature both the difficulties of nonlinear optimal control, as well as the range of issues introduced by discrete switching between modes of a hybrid system (this can include discontinuities in the vector field, reset of the continuous state, or even changes in the continuous state dimension due to algebraic constraints). This then motivates the study of subclasses of hybrid systems for which approximate solutions to reachability problems can be obtained.

1.4 Consideration of Subclasses of Hybrid Systems

In this dissertation, we will be specifically interested in controller design methods for the classes of continuous time switched nonlinear systems (Part I) and discrete time stochastic hybrid systems (Part II). The motivations for studying these subclasses of hybrid systems are discussed below.

A switched nonlinear system is a hybrid system whose continuous dynamics switches among a finite collection of continuous vector fields according to a discrete transition rule (Liberzon and Morse, 1999). The salient characteristics of such a system are: 1) the continuous state dimension is the same across the discrete modes; 2) the continuous state does not reset as a discrete transition is made. However, the vector field is in general discontinuous across a discrete transition. For

purposes of controller synthesis, we will consider the class of switched systems with *controlled switching* among the set of discrete modes. Within each discrete mode, the vector field can be nonlinear and subject to continuous disturbances with bounded magnitude. The underlying modeling assumption is that the condition for switching between the discrete modes is a design parameter, rather than an inherent characteristic of the physical system. This removes a number of technical difficulties associated with the analysis and control of continuous time hybrid systems, at the cost of restricting system dynamics to those without autonomous switches. However, a switched system model provides a fitting abstraction of a physical system whose underlying continuous behavior can be described in terms of a nonlinear vector field (up to bounded disturbances), but whose high level control is performed by discrete switching among a finite set of low level controllers. Examples of such systems can be found in aircraft conflict resolution (Tomlin et al., 2002), unmanned aerial vehicle trajectory control (Frazzoli et al., 2000), and robot formation control (Fierro et al., 2001).

As compared with deterministic hybrid systems, which takes switched nonlinear systems as a special case, stochastic hybrid system models allow for a probabilistic description of the uncertainties affecting system dynamics (Hu et al., 2000). This description can be obtained for example from a statistical analysis of past data on system behavior. Such models have been used to study control problems arising in air traffic management (Glover and Lygeros, 2004), communication networks (Hespanha, 2004), and systems biology (Hu et al., 2004). In the case of a discrete time stochastic hybrid system, the evolution of the system state over discrete time instants is assumed to be described by a transition probability over the hybrid state space, parameterized by the current system state and control inputs (Abate et al., 2008). This results in a discrete time Markov process, for which a rich body of theory has been built up in the study of stochastic optimal control (see for example Bertsekas and Shreve, 1978; Kumar and Varaiya, 1986). Under a discrete time model, there is no longer the notion of "continuity in time." As a result, one can account for discrete transitions which are dependent on the continuous state, possible changes in continuous state dimensions across discrete transitions, as well as resets in the continuous state, without significant technical difficulties. However, this level of generality comes at the cost of abstracting away the possibly rich set of hybrid system behaviors in between the discrete time instants.

As fields of study, the reachability of deterministic hybrid systems and stochastic hybrid systems are currently at two significantly different stages of development. Due to the large body of previous work which has focused on the modeling and analysis of deterministic hybrid systems, the former has by now a significant number of methods and algorithms for the computation of approximate reachable sets for a wide range of hybrid systems, with well-understood theoretical and numerical properties (see for example Asarin et al., 2000*a*; Kurzhanski and Varaiya, 2000; Bemporad et al., 2000*b*; Aubin et al., 2002; Chutinan and Krogh, 2003; Mitchell et al., 2005; Girard, 2005). On the other hand, the latter is still at a stage in which methods for formulating probabilistic reachability problems are in the process of being proposed (Koutsoukos and Riley, 2006; Amin et al., 2006), algorithms for computing approximations to the reachability probability are in the process of being devised (Hu et al., 2005; Abate et al., 2008), and their theoretical and numerical properties are in the process of being analyzed (Abate et al., 2010).

Given the gap between current understanding of deterministic and probabilistic reachability of

hybrid systems, the discussions in Part I and Part II of this dissertation will correspondingly differ in their focus with respect to aspects of the controller design problem. More specifically, our discussion of reachability problems for switched nonlinear systems will focus on the derivation of concrete design procedures and synthesis algorithms for generating feedback controllers that can be implemented in practical applications, based upon existing techniques for computing reachable sets. In comparison, our discussion of reachability problems for stochastic hybrid systems will be somewhat more abstract, and focus instead on the formulation of probabilistic reachability problems under various models of uncertainty, the construction of dynamic programming algorithms to solve these problems, and a foundational understanding of the theoretical properties and practical implications of the dynamic programming solution.

1.5 Organization

This dissertation covers several controller design methods for reachability problems that arise in the context of switched nonlinear systems and discrete time stochastic hybrid systems. Parts of the material presented here have appeared previously in several papers: Ding et al. (2008); Ding and Tomlin (2010); Ding et al. (2011*b*,*a*); Kamgarpour et al. (2011); Ding et al. (2012). In the following, we provide an overview of the main themes from each of the subsequent chapters.

In chapter 2, we consider switched nonlinear systems whose discrete states represent the sequential phases of a dynamic process. Within this context, a systematic procedure is proposed, based upon a hybrid system formalism, for carrying out controller design to satisfy sequential reachability specifications, namely specifications consisting of a sequence of safety and reach-avoid objectives. This is motivated by maneuver sequence design problems for unmanned aerial vehicles (UAVs) requiring robust operation guarantees. Through an appropriate choice of switching policy, the problem is posed as one of continuous control design for each discrete mode to ensure both individual reachability objectives, as well as proper composition between successive modes in the sequence. This design task is addressed using computational tools from nonlinear reachability analysis. The proposed methodology is illustrated through the example of automated aerial refueling (AAR).

In chapter 3, we shift the discussion to switched systems whose discrete states represent the set of qualitative control choices available to a high level controller. For this class of systems, controller synthesis algorithms are proposed for computing feedback control policies satisfying safety and reach-avoid specifications under worst-case disturbance realizations. For practical purposes, the problem is posed in a sampled-data setting in which measurements of the system state are obtained at regular sampling instants. The controller synthesis algorithms proceed by iterative reachability calculations over sampling intervals, returning as output a collection of reachable sets representing the control policy. These reachable sets can be then stored as lookup tables for online computation of control inputs in a sampled data setting. This methodology is applied to a simulation example of aircraft conflict resolution, as well as an experimental example of quadrotor hover control. The AAR example is also revisited to illustrate how the controller synthesis procedures can be applied to the design of switching controllers for individual phases of a sequential

reachability problem.

In chapter 4, we describe a framework for analyzing probabilistic reachability problems for discrete time stochastic hybrid systems (DTSHS) within a dynamic games setting. In particular, we consider zero-sum stochastic game formulations of the safety and reach-avoid problems, and discuss dynamic programming algorithms for computing the optimal probability of satisfying the reachability specifications, subject to the worst-case behavior of a rational adversary. This is motivated by instances of hybrid system models which feature a combination of stochastic and adversarial uncertainties. The problem is first posed in the finite horizon case, assuming an asymmetric information pattern favoring the adversary (as motivated by robust control problems). The implications of considering infinite horizon problems, as well as stochastic game formulations with symmetric information patterns are discussed in subsequent sections. In particular, the existence of a value in a symmetric stochastic game in general requires the consideration of randomized control policies.

In chapter 5, we focus our attention on the issue of partial observability in probabilistic reachability problems. We proceed by formulating the safety and reach-avoid problems for DTSHS as stochastic optimal control problems under partial observation, and show that even though they feature a multiplicative cost structure, they are equivalent to additive cost problems when the state space is augmented with an auxiliary binary random variable. This allows us to derive a sufficient statistic for probabilistic reachability problems as a probability distribution evolving on the augmented state space, as well as an abstract dynamic programming algorithm for computing the optimal probability of satisfying the reachability specifications. Issues of computation and implementation are discussed in terms of the special cases of finite state Markov decision processes and hybrid state models with probability density descriptions. In particular, practical implementation of the control and estimation algorithms hinges on efficient representations of the augmented probability distribution, which suggests the need for a deeper understanding of hybrid state estimation.

In chapter 6, we close with a summary of the main results presented in the dissertation, as well as some thoughts on directions on future work. These directions include investigations into computationally efficient methods for deterministic and probabilistic reachability, extensions of the controller synthesis methods for switched systems to handle autonomous switching, and the consideration of multi-objective problems and temporal objectives in reachability specifications. Most of the technical results and proofs in this dissertation are embedded within the main text, as they sometimes provide insight into particular aspects of the controller design procedure. However, some of the lengthy proofs related to measurability issues in Part II of the dissertation can be found in the appendices.

Part I

Switched Nonlinear Systems

Chapter 2

Design Procedure for Sequential Reachability Specifications

2.1 Motivation and Overview of Design Methodology

This chapter discusses a controller design procedure for sequential reachability specifications in the context of switched nonlinear systems. In particular, we consider switched systems whose discrete mode transitions follow a predefined sequence, and within each discrete mode, the objective is to satisfy either a safety or reach-avoid specification defined in terms of the continuous system trajectory, possibly subject to bounded continuous disturbances. The sequential structure of the system model provides an abstraction for dynamic processes whose flow follows a temporally-ordered sequence of qualitative phases, for example certain manufacturing or chemical processes. The reachability specification is then a description of the control objective in each phase of the process flow. Our primary motivation for studying such problems comes from automation of flight maneuver sequences for unmanned aerial vehicles (UAVs).

2.1.1 Automation of Flight Maneuver Sequences

In modern autonomous flight systems, the tasks of management and control of aircraft are frequently distributed between an onboard autonomous controller and external human operators or supervisors. In safety-critical scenarios, high level decisions on how and when flight maneuvers should be carried out currently rest almost exclusively with trained human operators, while the task of ensuring low level specifications such as flight envelope protection is delegated to the onboard flight control system (Yavrucuk et al., 2009). However, as one pushes towards increased levels of autonomy for UAV operation, it becomes important to investigate methods for incorporating some of the high level decision making capabilities into the onboard UAV control system.

In this chapter, we will restrict our attention to high level specifications consisting of an ordered sequence of waypoints that the UAV must reach, while satisfying a safety constraint, for example avoiding a collision with another aircraft. Following this specification, one can separate the control task into a sequence of qualitative phases, with each phase corresponding to a flight ma-

neuver in which the objective is to either reach a target waypoint in finite time while satisfying a safety constraint (i.e. reach-avoid problem), or to loiter within a neighborhood of the waypoint (i.e. safety/invariance problem). The transitions between maneuvers can be controlled by human operators or initiated autonomously by the UAV. This then results in a sequential reachability problem. In the following, we discuss a practical example of such type of specifications.

The scenario is that of Automated Aerial Refueling (AAR). Currently, manned military aircraft which undergo long range missions are routinely refueled in mid-air by tanker aircraft. As the use of UAVs becomes increasingly prevalent, there is an ongoing effort to introduce this capability into UAV operations, ideally with a minimal amount of supervision by human operators (see for example Valasek et al., 2002; Nalepka and Hinchman, 2005; Jin et al., 2006; Ross et al., 2006). A conceptual illustration of the AAR scenario is shown in Figure 2.1.

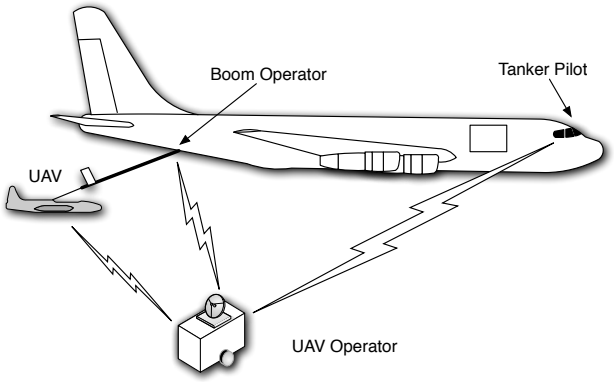


Figure 2.1: Conceptual illustration of automated aerial refueling scenario.

During a refueling operation, a UAV detaches from its formation, and approaches the rear of a tanker aircraft for refueling. The boom operator onboard the tanker then lowers a fuel boom to refuel the UAV; once the refueling is complete, the operator disconnects the boom and the UAV breaks away from the tanker to rejoin its formation. This description naturally decomposes AAR into several distinct phases, namely an "approach tanker" phase, a "refueling" phase, and a "rejoin formation" phase. To introduce further structure into the refueling operation, the approach and rejoin phases can be separated into a sequence of flight maneuvers in which the objective is to reach some target location relative to the tanker aircraft, while avoiding collisions. The AAR scenario can be then modeled by a switched system whose discrete dynamics consist of the sequential transitions through the flight maneuvers, and the continuous dynamics consist of the relative kinematics between the tanker and the UAV in executing the respective maneuvers. The controller design problem for this scenario can be then posed as a sequential reachability problem in which the objective of each maneuver is to reach a waypoint location relative to the tanker aircraft, while avoiding a collision, namely a reach-avoid objective. If one were to consider possible environment disturbances such as wind effects or variations in tanker speed, then this specification would need to be satisfied subject to the worst-case realizations of the disturbances.

The scenario described above is in large part that of an autonomous AAR procedure. Namely, except for the initiation of the refueling operation and the actual refueling of the UAV by the boom operator, the execution of the maneuver sequence does not require any additional human intervention. However, there may be cases in which either the UAV operator or boom operator would like to have an input on when the UAV is allowed to transition from one maneuver to the next. This could arise for example from the need to interrupt the refueling sequence due to severe wind turbulences beyond those considered for the autonomous AAR design, or to have the UAV dwell within the vicinity of the fuel boom while the boom operator refuels the UAV. For such cases, one can insert intermediate maneuvers into the refueling sequence with the objective of keeping the UAV in a neighborhood of each waypoint while awaiting operator confirmation to perform the next maneuver. The resulting reachability problem then consists of a sequence of reach-avoid and safety/invariance objectives.

The development of the AAR scenario for this research effort was carried out in conjunction with Boeing Research & Technology, and the Air Force Research Laboratory (AFRL), through the Certification Technologies for Flight Critical Systems (CerTA FCS) project. The author would like to gratefully acknowledge the contributions of Jim Barhorst, Jim Paunicka, and Doug Stuart of Boeing Research & Technology for their valuable feedback and suggestions in formulating the AAR scenario. Also, many of the research ideas which led to the work described in this chapter were conceived in David Homan's yearly meetings on Verification and Validation at Wright-Patterson Air Force Base in Dayton, OH. In these meetings, our effort was influenced by the conversations and presentations of many of the participants, and the author is grateful for their contributions.

2.1.2 Methodology Overview

Our approach to the sequential reachability problem is to pose it as a hybrid system design problem. Within this framework, the design parameters include the switching conditions for the sequence of discrete modes, as well as the continuous control law within each of the discrete modes. As will be discussed, through a judicious choice of switching condition, one can isolate the problem to one of continuous control design for the individual discrete modes. In particular, the continuous control law in each mode needs to satisfy

1. the reachability specification given for that mode;

2. a compatibility condition to ensure that the sequence of modes can be properly composed.

It turns out that satisfying these requirements can be formulated as a continuous time reachability problem. As such, we can use computational reachability analysis for continuous time systems as a design tool for the continuous control laws. This allows us to check whether a given control law satisfies the desired reachability specifications without the need to resort to exhaustive simulation studies. Due to our consideration of nonlinear continuous dynamics subject to continuous disturbance, the reachability calculations will be carried out using a method based upon numerical solutions of Hamilton-Jacobi partial differential equations (PDEs) (Mitchell et al., 2005).

To discuss this in a more concrete setting, consider again the aerial refueling scenario. In the case that the sequence of refueling maneuvers is to be performed autonomously, a reasonable choice of switching conditions is to specify that as soon as the UAV reach a given waypoint, it will immediately transition to the next maneuver. A computational reachability analysis can then be performed for each flight maneuver in the refueling sequence to determine 1) the *capture set*: the set of aircraft states from which a maneuver can be completed within a finite time horizon; and 2) the *collision set*: the set of aircraft states from which the trajectory of a flight maneuver passes through a collision zone centered on the tanker aircraft. At design time, the capture sets and collision sets computed for the various maneuvers in the AAR sequence can be used to guide the choice of maneuver control laws so as to ensure that each maneuver terminate in an aircraft state which satisfies the reach-avoid objective of the next maneuver (thus allowing the next maneuver to be feasibly initiated). Furthermore, through appropriate modifications of the reachability analysis, the effects of bounded environment disturbances can also be taken into account. However, in such cases, the resulting design of maneuver control laws is in general more conservative than the case in which the robustness factors are not considered.

2.1.3 Organization

The organization of this chapter is as follows. Section 2.2 provides an overview of related work in the domain of formal verification and mode sequence design. Section 2.3 discusses a hybrid formalism for the class of switched systems under consideration. Section 2.4 provides formal statements of two types of sequential reachability problems. Section 2.5 briefly reviews the method of Hamilton-Jacobi reachability for nonlinear continuous systems. Section 2.6 introduces a reachability-based procedure for performing controller design to satisfy sequential reachability specifications. Section 2.7 discusses the use of reachable sets as an aid to human decision making in recovering from a class of fault conditions occurring during run-time, in particular that of improper initialization. These methods are then specialized to the particular case of automated aerial refueling, and the results of the controller design procedure along with simulated scenarios are presented in section 2.8.

2.2 Related Work

2.2.1 Hamilton-Jacobi Reachability

The method of Hamilton-Jacobi (H-J) reachability is developed for computing reachable sets for continuous time nonlinear system, under a dynamic games framework (Mitchell et al., 2005). In the work by Tomlin et al. (2003), one can find a comprehensive overview of the computational techniques underlying the H-J reachability, its use in analyzing and verifying continuous time nonlinear systems as well as hybrid systems.

This method has seen successes in numerous aeronautical applications. In the work by Mitchell et al. (2005), the authors present a method for detecting possible "loss of separation" between pairs

of aircraft over a given airspace, based upon backward reachable sets computed using H-J PDEs. Using this formulation of the collision avoidance problem, the reachable set method has been used to verify safety of conflict resolution aircraft maneuvers (Tomlin et al., 2001), and closely spaced parallel approaches for airport runways (Teo and Tomlin, 2003). The results of the reachability calculations were validated in extensive simulations as well as UAV flight experiments (Jang and Tomlin, 2005; Teo, 2005). While the focus of these previous applications lies largely in safety verification, the work described in this chapter proposes a method for using reachability analysis as a design tool for choosing the continuous control laws of a maneuver sequence so as to satisfy the desired specifications.

In systems that involve human-automation interactions, H-J reachability has also been successfully demonstrated as a method for informing human decisions. In the work by Oishi et al. (2002), the authors use reachability analysis to determine whether the pilot display of a civil jet aircraft contained enough information for the pilot to safely perform a Take-off/Go-Around (TO/GA) maneuver from a Flare landing maneuver. In another example, as described in Sprinkle et al. (2005), reachable sets computed using H-J methods are used to inform decisions on the re-initiation of a landing maneuver during TO/GA, and the results were demonstrated on a fixed-wing UAV (T-33). Building upon these previous works, this chapter also discusses an approach for using reachable sets as a visual tool for guiding human operator decisions in the scenario that a maneuver sequence is improperly initialized.

2.2.2 Alternative Reachability Approaches

Aside from H-J reachability, there is a myriad of alternative approaches in the domain of reachable set based system verification for hybrid systems. The work considering timed automata and linear hybrid automata includes seminal papers by Alur and Dill (1994) and Henzinger (1996). Results have been generalized to linear and nonlinear continuous dynamics, with supporting computational tools (Asarin et al., 2000*a*; Botchkarev and Tripakis, 2000; Kurzhanski and Varaiya, 2000; Bemporad et al., 2000*b*; Aubin et al., 2002; Chutinan and Krogh, 2003; Girard, 2005; Han and Krogh, 2006). Methods that operate on system abstractions can reduce computational complexity, including simulation and bisimulation relations (Alur et al., 2000; Haghverdi et al., 2005; Girard et al., 2008), which are used to construct discrete abstractions of hybrid system dynamics. In comparison, the H-J method has the advantage of being able to handle non-convex sets, nonlinear continuous dynamics, and differential games, while providing subgrid accuracy using implementations of the level set methods (Sethian, 1999; Osher and Fedkiw, 2002). Furthermore, it is versatile with respect to the range of reachable set computations that can be performed. This includes computations under forward propagation or backward propagation in time, with either existentially quantified (i.e. reach for some input) or universally quantified inputs (i.e. reach for all inputs). This feature becomes important when one would like to perform reachability computation under disturbances, as capture set computation would require universally quantified disturbance inputs, while unsafe set computation would require existentially quantified disturbance inputs. The benefits of the H-J method, however, comes at the cost of higher computational complexity with respect to some of the alternative reachability methods.

In reachability work relating to stochastic systems, Prandini and Hu (2006) discuss the use of Markov chains to determine the reachability of some stochastic system in some lookahead time (potentially infinite). Air traffic management as a driving example for distributed control and stochastic analysis of safety-critical real-time systems is demonstrated in the HYBRIDGE report (Blom and Lygeros, 2005). In many of these applications, events that jeopardize the safety of the system are rare, and using probabilistic methods such as Monte Carlo simulations (Blom et al., 2007), it is possible to estimate the probability of these events through stochastic reachability and obtain some measure of confidence in the safety of a system design (Blom et al., 2009). On the other hand, for systems with environment disturbances that are known to lie within certain bounds, deterministic reachability can be used to provide stronger performance guarantees for relevant disturbances such as perturbations in velocity or heading.

2.2.3 Flight Maneuver Design Approaches

State feedback is a common approach to the design and implementation of flight maneuvers. In general, a trajectory is generated (or designed) and the vehicle tracks this trajectory based on an on-board guidance and navigation system. Depending on the maneuver, this trajectory may be globally fixed (for example, a glideslope for landing) or defined from a location decided at flight time (for example, a waypoint). For certain maneuvers, additional scrutiny is given due to their proximity to regions of stall or other vehicles. Details for optimal Go-Around and Flare maneuvers are given in the work of Buell and Leondes (1973). Interestingly, transitions between these maneuvers can also be discussed in the framework of reachability, as in the previously mentioned work by Oishi et al. (2002).

Alternatively, maneuver sequence synthesis may be performed at runtime using path-planning algorithms. In the work by Bottasso et al. (2008), the authors demonstrate smooth path planning using motion primitives to pass through a series of waypoints constituting a track. This approach is related to that applied by Frazzoli et al. (2005) and Koo et al. (2001), both of which are focused on rotorcraft. Although these algorithms are computationally efficient, providing robust performance guarantees are often complicated by the presence of model uncertainty and environment disturbances at runtime.

To address safety concerns, safe maneuvers with real-time trajectory generation were shown by Waydo et al. (2007) for the case of formation flight with an autonomous vehicle, where several control modes are used depending on loss of communication with a manned vehicle. This approach was proved safe using runtime predictive control, requiring a solution to the stationary Riccati equation over (essentially) infinite time. It is interesting to compare this approach to backward reachability, as it can be essentially thought of as a forward reachability calculation to validate a specific trajectory (rather than validate all potential trajectories using backward reachability).

As an alternative, Lyapunov functions can be also used to provide robust guarantees on the closed-loop performance of the system under a given controller design. Relevant to the work in this chapter is a Lyapunov-based method proposed by Burridge et al. (1999), in the context of motion planning applications, for composing sequences of local feedback controllers to achieve a desired final configuration. Under this method, the authors construct local controllers whose domains of

attraction are estimated from the level sets of Lyapunov functions. Sequential composition is then performed by ensuring that the goal set of a given controller is contained within the domain of attraction of the next controller in the sequence.

For applications with nonlinear continuous dynamics, constructing appropriate Lyapunov functions satisfying the desired stability objectives can be a non-trivial task. Depending on the choice of Lyapunov functions, estimates of the domain of attraction can be also quite conservative, especially when system dynamics are perturbed by disturbances. By using H-J methods to generate the relevant reachable sets, the methodology proposed in this chapter avoids the need for selecting Lyapunov functions, while reducing the conservatism in estimating the domain of attraction. Also, it is worth noting that local controllers produced through Lyapunov methods can be evaluated using Hamilton-Jacobi reachability for satisfaction of target attainability and safety objectives. Thus, the presented approach is not meant to supplant existing methods for robust nonlinear controller design, but to augment them.

2.3 Hybrid Model of Sequential Transition Systems

In this section, we will introduce the necessary modeling formalism for the controller design problem. In particular, the focus will be on *sequential transition systems*, which are switched nonlinear systems whose discrete mode transitions follows a pre-defined sequence. As preliminaries, we will first review a general hybrid system model, based upon the formalism described in Lygeros et al. (1999*b*) and Tomlin et al. (2000). Sequential transition systems will be then discussed as instantiations of this general model.

2.3.1 General Hybrid Automaton

Over the several decades of research on hybrid systems, numerous modeling frameworks have been introduced in literature. For our purposes, the formalisms proposed in Lygeros et al. (1999*b*) and Tomlin et al. (2000) provides a sufficiently rich class of models to describe the behavior of sequential transition systems. The description given below is correspondingly adapted from these previous works.

Definition 2.1 (Hybrid Automaton)**.** A hybrid automaton is a tuple

$$\mathscr{H} = (Q, X, \Sigma, V, Init, f, Dom, Reset),$$

defined as follows.

- *Discrete state space* $Q := \{q_1, q_2, ..., q_m\}$, $m \in \mathbb{N}$.
- *Continuous state space* $X := \mathbb{R}^n$, , $n \in \mathbb{N}$.
- *Discrete input space* $\Sigma := \Sigma_1 \times \Sigma_2$, where Σ_1 is the set of discrete control inputs and Σ_2 is the set of discrete disturbance inputs.

- *Continuous input space* $In := U \times D$, where U is the set of continuous control inputs and D is the set of continuous disturbance inputs.
- *Admissible initial conditions* $Init \subseteq Q \times X$.
- *Vector field* $f : Q \times X \times In \to X$, describing the continuous state evolution.
- *Domain* $Dom \subseteq Q \times X \times \Sigma \times In$, describing the domain on which continuous state evolution is permitted.
- *Reset relation* $Reset : Q \times X \times \Sigma \times In \to 2^{Q \times X}$, describing the subset of the hybrid state space that the system state is permitted to transition to in the event of a discrete jump.

In order to ensure the existence and uniqueness of continuous trajectory under the vector field f, we will need f to satisfy certain regularity assumptions.

Assumption 2.1. The vector field f is continuous and bounded, and that for fixed $q \in Q$, $(u,d) \in In$, the function $x \to f(q,x,u,d)$ is Lipschitz continuous.

Roughly speaking, an execution of the hybrid automaton proceeds as follows. From an initial condition $(q_0,x_0) \in Init$, the continuous trajectory $x(\cdot)$ evolves according to the ordinary differential equation $\dot{x} = f(q_0,x,u,d), x(0) = x_0$, while the discrete state remains constant, as long as $(q_0,x(t),\sigma_1(t),\sigma_2(t),u(t),d(t)) \in Dom$. At the first time instant t_1 when this condition no longer holds, the system state takes a discrete jump to

$$(q',x') \in Reset(q_0,x(t_1),\sigma_1(t_1),\sigma_2(t_1),u(t_1),d(t_1)),$$

and the cycle repeats. To define this more formally, we will need the notion of a hybrid time trajectory.

Definition 2.2 (Hybrid Time Trajectory (Lygeros et al., 1999*b*; Tomlin et al., 2000))**.** A hybrid time trajectory $\tau = \{L_i\}_{i=1}^{N}$ is a finite ($N < \infty$) or infinite ($N = \infty$) sequence of intervals of the real line satisfying:

- $L_k = [\tau_k, \tau_k']$, $k < N$, and if $N < \infty$, either $L_N = [\tau_N, \tau_N']$ or $L_N = [\tau_N, \tau_N')$;
- $\forall k = 1,...,N$, $\tau_k \leq \tau_k' = \tau_{k+1}$.

As in Lygeros et al. (1999*b*) and Tomlin et al. (2000), for a given $t \in \mathbb{R}$ and a hybrid time trajectory τ, we will use $t \in \tau$ to refer to $t \in L_k$ for some $k = 1,...,N$. Informally, a hybrid time trajectory is a sequence of time intervals on which continuous evolution takes place. However, there may be cases in which another discrete jump takes place immediately after a discrete jump. In such cases, there could be time intervals L_k with measure zero. We can now give a definition for the execution of a hybrid system.

Definition 2.3 (Execution of Hybrid Automaton (Lygeros et al., 1999*b*; Tomlin et al., 2000))**.** An execution of a hybrid automaton $\mathscr{H}$ is a collection $\chi = (\tau, q, x, \sigma, u, d)$ with $(q,x) : \tau \to Q \times X$ and $(\sigma, u, d) : \tau \to \Sigma \times In$ satisfying:

- $(q(\tau_0), x(\tau_0)) \in Init$;
- $\forall i = 1, ..., N$, $(q(\tau_{k+1}), x(\tau_{k+1})) \in Reset(q(\tau'_k), x(\tau'_k), \sigma(\tau'_k), u(\tau'_k), d(\tau'_k))$;
- On every interval $L_k = [\tau_k, \tau'_k]$ such that $\tau_k < \tau'_k$, $q(\cdot)$ and $\sigma(\cdot)$ are constant, $v(\cdot)$ is piecewise continuous, $x(\cdot)$ is the solution to $\dot{x} = f(q, x, u, d)$, and $(q(t), x(t), \sigma(t), u(t), d(t)) \in Dom$, $\forall t \in L_k$.

An execution $\chi = (\tau, q, x, \sigma, u, d)$ is said to be *finite* if τ is a finite sequence of time intervals with the last interval being closed. An execution $\chi = (\tau, q, x, \sigma, u, d)$ is said to be *infinite* if either τ is an infinite sequence of time intervals, or if $\sum_{k=1}^{N} (\tau'_k - \tau_k) = \infty$.

From Definition 2.3, it can be seen that the reset relation *Reset* specifies conditions under which discrete jumps are enabled. Namely, for a given state $(q,x) \in Q \times X$ and an input $(\sigma, u, d) \in \Sigma \times In$, if the set $Reset(q, x, \sigma, u, d)$ is nonempty, then a discrete jump is permitted and the jump can be taken to any state in $Reset(q, x, \sigma, u, d)$. However, one may also choose not to take the jump, as long as $(q, x, \sigma, u, d) \in Dom$ is satisfied, namely the state and input lies in the domain on which continuous evolution is allowed. Thus, it can be seen that there is a degree of nondeterminism in the executions of a hybrid automaton. In particular, for a given hybrid automaton $\mathscr{H}$, initial condition $(q_0, x_0) \in Init$, and inputs $\sigma(\cdot)$, $u(\cdot)$, $d(\cdot)$, there may not exist, in general, an infinite execution, and if one exists, it may not be unique. A complete discussion of the issue of existence and uniqueness is somewhat involved for a general hybrid automaton. The interested reader is referred to the work of Lygeros et al. (1999*a*), in which some sufficient conditions for the existence and uniqueness of infinite executions are given.

2.3.2 Automated Sequential Transition Systems

We first develop a model for sequential transition systems in which there are no external discrete inputs, namely $\Sigma_2 = \emptyset$. Through an interpretation of Σ_2 as the set of commands issued by an external human operator, this corresponds to a system operating under automated selections of control inputs $\sigma_1 \in \Sigma_1$ and $u \in U$. In other words, once the system has been initialized within the initial set *Init*, the system would proceed through the sequence of operating modes without further intervention from the human operator. However, in the execution of this mode sequence, the continuous trajectory may still be perturbed by environment disturbances, which are modeled as continuous disturbance inputs. We will refer to this class of systems as *automated sequential transition systems*.

Definition 2.4 (Automated Sequential Transition System)**.** An automated sequential transition system is a hybrid automaton $\mathscr{H}$ with

- *Switching control space*: $\Sigma = \Sigma_1 = \{\sigma_1, \sigma_2, ..., \sigma_m\}$;

- *Initial conditions*: $Init = q_1 \times X_0$, with $X_0 \subseteq X$;
- *Mode domains*: $Dom = \bigcup_{i=1}^{m} q_i \times X \times \Sigma^i \times In$, where $\Sigma^i = \Sigma \setminus \{\sigma_{i+1}\}$ for $i = 1, ..., m-1$ and $\Sigma^m = \Sigma$;
- *Automated switches*: For every $i = 1, ..., m-1$, $x \in X$, and $(u,d) \in In$, $Reset(q_i, x, \sigma, u, d) = (q_{i+1}, x)$ if $\sigma = \sigma_{i+1}$, and $Reset(q_i, x, \sigma, u, d) = \emptyset$ otherwise; for $i = m$, $Reset(q_m, x, \sigma, u, d) = \emptyset$, $\forall x \in X$, $\sigma \in \Sigma$, $(u,d) \in In$.

The above definition describes a hybrid system in which the set of discrete inputs are the switching controls for transitions between the successive discrete modes. It can be verified using the conditions given in Lygeros et al. (1999*a*) that for any initial condition $(q_0, x_0) \in Init$ and any measurable realizations of the input signals $\sigma(\cdot)$, $u(\cdot)$, $d(\cdot)$, there exists a unique infinite execution for this system. This execution proceeds as follows. The system state is initialized in the first discrete mode q_1 within a set X_0. The continuous trajectory evolves according to the continuous dynamics in q_1 until a switching input σ_2 to transition to q_2 is applied. A discrete jump is then taken to q_2 and the continuous trajectory evolves according to the continuous dynamics in q_2. This proceeds until the discrete trajectory reaches mode q_m, upon which time the trajectory evolves according to the dynamics in q_m without any further discrete transitions.

Due to the presence of continuous disturbances, controller design for this class of systems will be carried out in terms of feedback control policies. In particular, we will consider switching control policies of the form $F : Q \times X \to \Sigma_1$ such that $\sigma(t) = F(q(t), x(t))$, $\forall t \geq 0$, and continuous control policies of the form $K : Q \times X \to U$ such that $u(t) = K(q(t), x(t))$, $\forall t \geq 0$. Taken together, a control policy (F, K) is said to be admissible if for any measurable realizations of the disturbance d, there exists a unique infinite execution for the closed-loop system.

2.3.3 Semi-Automated Sequential Transition Systems

Instead of a fully automated system, there may be cases in which one would want to allow for a degree of human intervention in order to guard against contingencies during system operation. Here we will consider interventions in the form of commands determining if and when the system should proceed to the next task in the mode sequence. This is a simple example of a *mixed-initiative* system, in which the system is subject to both human and automated decisions. Such a type of system is a subject of ongoing research on human-UAV interactions (Lam et al. (2008); Cummings and Mitchell (2008)).

Our approach to modeling this interaction is to interpret the discrete control space Σ_2 as the set of human operator commands which provides confirmation that the system can proceed to the next phase of operation. Intermediate operating modes are then introduced into the mode sequence, with outgoing transitions that are specifically controlled by these commands. We refer to this as a *semi-automated sequential transition system*.

Definition 2.5 (Semi-Automated Sequential Transition System)**.** A semi-automated sequential transition system is a hybrid automaton $\mathscr{H}$ with

- *Discrete state space*: $Q = \{q_1, \hat{q}_1, q_2, \hat{q}_2, ..., q_m, \hat{q}_m\}$;
- *Switching control space*: $\Sigma = \Sigma_1 \times \Sigma_2$, where $\Sigma_1 = \{\sigma_1, \sigma_2, ..., \sigma_m\}$ and $\Sigma_2 = \{\hat{\sigma}_1, \hat{\sigma}_2, ..., \hat{\sigma}_m\}$;
- *Initial conditions*: $Init = q_1 \times X_0$, with $X_0 \subseteq X$;
- *Mode domains*: $Dom = \bigcup_{i=1}^{m} \left(q_i \times X \times \Sigma^i \times In\right) \cup \left(\hat{q}_i \times X \times \hat{\Sigma}^i \times In\right)$, where $\Sigma^i = (\Sigma_1 \setminus \{\sigma_{i+1}\}) \times \Sigma_2$ for $i = 1, ..., m$, and $\hat{\Sigma}^i = \Sigma_1 \times (\Sigma_2 \setminus \{\hat{\sigma}_{i+1}\})$ for $i = 1, ..., m-1$ and $\hat{\Sigma}^m = \Sigma$;
- *Automated Switches*: For every $i = 1, ..., m$, $x \in X$, and $(u,d) \in In$, $Reset(q_i, x, \sigma, u, d) = (\hat{q}_i, x)$ if $\sigma = (\sigma_{i+1}, \hat{\sigma})$ for some $\hat{\sigma} \in \Sigma_2$, and $Reset(q_i, x, \sigma, u, d) = \emptyset$ otherwise;
- *Externally Controlled Switches*: For every $i = 1, ..., m-1$, $x \in X$, and $(u,d) \in In$, $Reset(\hat{q}_i, x, \sigma, u, d) = (q_{i+1}, x)$ if $\sigma = (\sigma, \hat{\sigma}_{i+1})$ for some $\sigma \in \Sigma_1$, and $Reset(\hat{q}_i, x, \sigma, u, d) = \emptyset$ otherwise; for $i = m$, $Reset(\hat{q}_m, x, \sigma, u, d) = \emptyset$, $\forall x \in X$, $\sigma \in \Sigma$, $(u,d) \in In$.

In the above definition, the discrete states $\{q_i\}_{i=1}^m$ can be interpreted as the modes in which the task specifications of the sequential transition system are carried out. We refer to them as *transition states*. On the other hand, the discrete states $\{\hat{q}_i\}_{i=1}^m$ can be interpreted as the modes in which the system awaits confirmation by the operator to proceed to the next task. We refer to them as *stationary states*. From the point of view of the operator, the operation of a semi-autonomous system involves a sequence of phases in which the automation carries out a task in a transition state and then pauses for further instruction in a stationary state.

Similarly as in the automated case, one can verify using the conditions given in Lygeros et al. (1999*a*) that for any initial condition $(q_0, x_0) \in Init$ and any measurable realizations of the input signals $\sigma(\cdot)$, $u(\cdot)$, $d(\cdot)$, there exists a unique infinite execution for the semi-automated system. A more formal description of the system execution can be given as follows. The system state is initialized in the first discrete mode q_1 within a set X_0. The continuous trajectory evolves according to the continuous dynamics in q_1 until a switching input σ_2 is applied. At this time, a discrete jump is taken to $\hat{q}_1$ to wait for an external command. When the command $\hat{\sigma}_2$ is received to proceed to q_2, the system trajectory takes a discrete jump to q_2, and the continuous trajectory evolves in q_2 until a command σ_3 is received to transition to $\hat{q}_2$. This proceeds until the discrete trajectory reaches $\hat{q}_m$, upon which time the trajectory evolves according to the dynamics in $\hat{q}_m$ without any further discrete transitions.

Controller design for this class of systems also consists of a choice of switching policy $F : Q \times X \to \Sigma_1$, as well as a choice of continuous control policy $K : Q \times X \to U$. However, it should be noted that the choice of switching policy in the stationary states is largely irrelevant, as the outgoing transitions are controlled by external commands. A control policy (F, K) is said to be admissible if for any measurable realizations of the disturbance d, there exists a unique infinite execution for the closed-loop system.

2.4 Sequential Reachability Problems

2.4.1 Specification with Reach-avoid Objectives

First, consider the case of an automated sequential transition system and a problem specification in which the objective in each mode q_i is to drive the continuous state x into a desired target set $R_i \subset X$ within finite time, while avoiding a set $A_i \subset X$. Here the sets R_i could for example represent a sequence of waypoints, while the sets A_i could for example represent unsafe operating conditions or obstacles in the environment.

Problem 2.1 (Sequential Reachability Problem for Automated Transition System)**.** Given an automated sequential transition system $\mathscr{H}$, target sets $R_i \subset X$, $i = 1,...,m$, and avoid sets $A_i \subset X$, $i = 1,...,m$, choose an admissible control policy (F,K) such that for any measurable realization of the disturbance d satisfying $d(t) \in D$, $\forall t \geq 0$, the unique infinite execution of $\mathscr{H}$ satisfies

1. $(q(t_i),x(t_i)) \in q_i \times R_i$ for some sequence of times $t_0 = 0 \leq t_1 \leq \cdots \leq t_m < \infty$;
2. On any time interval $[\tau_k, \tau_k'] \in \tau$ such that $q(\cdot) \equiv q_i$, $x(t) \notin A_i$, $\forall t \in [\tau_k, \tau_k']$.

2.4.2 Specification with Reach-avoid and Invariance Objectives

Now we consider the case of a semi-automated sequential transition system. It is assumed that the objectives of the sequence of transition modes are still reach-avoid objectives. However, within the stationary modes, the objectives are of the invariance type, namely stay within a neighborhood of a target set, until a command is given to perform the next task in the sequence.

Problem 2.2 (Sequential Reachability Problem for Semi-Automated Transition System)**.** Given a semi-automated sequential transition system $\mathscr{H}$, target sets $R_i \subset X$, $i = 1,...,m$, avoid sets $A_i \subset X$, $i = 1,...,m$, and target neighborhoods $W_i \subset X$ satisfying $R_i \subseteq W_i \subset A_i^C$, choose an admissible control policy (F,K) such that for any measurable realization of the disturbance d satisfying $d(t) \in D$, $\forall t \geq 0$, and any measurable realization of the external switching command σ_2 satisfying $\sigma_2(t) \in \Sigma_2$, $\forall t \geq 0$, the unique infinite execution of $\mathscr{H}$ satisfies

1. If $q(t) = q_i$ for some $t \in \tau$, then there exists $t_i < \infty$ such that $(q(t_i),x(t_i)) \in q_i \times R_i$; furthermore, on any time interval $[\tau_k, \tau_k'] \in \tau$ such that $q(\cdot) \equiv q_i$, $x(t) \notin A_i$, $\forall t \in [\tau_k, \tau_k']$;
2. On any time interval $[\tau_k, \tau_k'] \in \tau$ such that $q(\cdot) \equiv \hat{q}_i$, $x(t) \in W_i$, $\forall t \in [\tau_k, \tau_k']$.

In other words, if the discrete trajectory reaches a transition state q_i, then the target set R_i is to be attained within finite time while staying away from the avoid set A_i. Moreover, whenever the discrete trajectory reaches a stationary state $\hat{q}_i$, the continuous trajectory remains within the target neighborhood W_i until the discrete trajectory jumps away from $\hat{q}_i$. However, there may be cases in which a command to switch to q_{i+1} is never received, and the state trajectory remains within $\hat{q}_i \times W_i$ indefinitely.

2.4.3 Formulation in Terms of Continuous Reachability Problems

As discussed in section 2.1, our approach to problems 2.1 and 2.2 is to choose an appropriate switching policy so as to reduce these problems to a sequence of continuous control design problems. In particular, given the sequential nature of these problems, as well as the fact that the objective of the system in each transition mode is to reach a target set in finite time, there is no reason for the system to dwell in a given transition mode once the reach-avoid objective is attained. Thus, a reasonable choice of switching policy is to transition to the next mode in the sequence once a target set in a transition mode is reached. Specifically, we consider a switching policy F satisfying

$$F(q_i,x) = \begin{cases} \sigma_{i+1}, & x \in R_i \\ \sigma_i, & \text{otherwise.} \end{cases} \tag{2.1}$$

for every transition mode q_i, $i = 1,...,m-1$. It turns out that this choice of switching policy is sufficient for Problem 2.1. However, a slight modification of the switching region will be needed to ensure the invariance objectives for Problem 2.2.

Now consider the problem of designing the continuous control policy K. From the problem descriptions, one can deduce certain requirements for the continuous control design. In particular, for the case of the automated sequential transition system, the continuous trajectories resulting from a control law in transition mode q_i should satisfy a reach-avoid objective, namely $x(t) \in R_i$ for some $t < \infty$, and $x(t') \notin A_i$ for every $t' \leq t$, over a subset of initial conditions in $\mathbb{R}^n$ which ensures proper composition with the continuous trajectories of the previous mode q_{i-1}. Given the choice of switching policy F, this set is simply given by the previous target set R_{i-1}. By solving this sequence of reach-avoid problems, we obtain a control policy K satisfying the specifications of problem 2.1.

In the case of the semi-automated sequential transition system, the specification also requires that the continuous trajectories in each stationary mode $\hat{q}_i$ satisfy an invariance objective, namely remain within a target neighborhood W_i for all time. The control law design then needs to account for the composition between transition and stationary modes. In particular, the control design for a transition mode q_i should ensure that the reach-avoid objective is achieved for every initial condition in the previous target neighborhood W_{i-1}, while the control design for a stationary mode $\hat{q}_i$ should ensure that the invariance objective is achieved for a subset of the target set R_i (which replaces the switching region in (2.1)). By solving this sequence of reach-avoid and invariance problems, we obtain a control policy K satisfying the specifications of problem 2.2.

2.5 Overview of Hamilton-Jacobi Reachability

By treating a sequential reachability problem as a sequence of continuous reachability problems, our controller design procedures for problems 2.1 and 2.2 will involve the use of continuous time reachability analysis as a design tool for the continuous control laws. In this section, we will review some basic forms of reachable sets, as well as a method for computing an approximation of these sets for nonlinear continuous time systems, based upon the numerical solution of an appropriate

Hamilton-Jacobi PDE (Mitchell et al., 2005). For the rest of this section, we will assume the following system dynamics.

$$\dot{x} = f(x,u,d),\ x(0) = x_0, \tag{2.2}$$

where $x \in \mathbb{R}^n$ is the continuous state, $u \in U$ is the control input, and $d \in D$ is the disturbance input. Here we assume that the sets U and D are compact. In order to the apply the computational procedure described in Mitchell et al. (2005) to our problem, the regularity assumptions on the vector field f needs to be slightly strengthened as compared with Assumption 2.1.

Assumption 2.2. The vector field f is uniformly continuous and bounded, and that for fixed $d \in D$, the function $(x,u) \to f(x,u,d)$ is Lipschitz continuous.

2.5.1 Capture Set

For a given target set $R \subset \mathbb{R}^n$, and time horizon $T \geq 0$, the capture set of (2.2) is the set of initial conditions x_0 for which there exists a choice of control strategy such that, regardless of the choice of disturbance strategy, there exists a time instant $t \in [0,T]$ such that $x(t) \in R$. If one were to view this as a zero-sum differential game (see for example Isaacs, 1967; Evans and Souganidis, 1984; Başar and Olsder, 1999) in which the objective of the control is to reach the set R within $[0,T]$, then this is the set of winning initial conditions for the control. A formal definition for this set requires some amount of notation and concepts from differential games (Mitchell et al., 2005). For our purposes, however, it is sufficient to consider a definition for the case in which the control strategy is fixed. As a notational convenience, we define the set of admissible disturbance realizations over a time interval $[0,T]$ as

$$\mathscr{D}_T = \{d : [0,T] \to D | d(\cdot) \text{ is measurable}\}.$$

Definition 2.6 (Capture Set)**.** Given a target set R, a time horizon T, and a Lipschitz continuous feedback law $K : \mathbb{R}^n \to U$, the capture set $\mathscr{R}(R,K,T)$ of (2.2) is given by

$$\mathscr{R}(R,K,T) = \{x_0 \in X : \forall d(\cdot) \in \mathscr{D}_T, \exists t \in [0,T],\ x(t) \in R\},$$

where $x(\cdot)$ is the solution of $\dot{x}(t) = f(x(t),K(x(t)),d(t)),\ x(0) = x_0$ on the interval $[0,T]$.

By fixing the continuous feedback law K, the problem of computing a capture set $\mathscr{R}(R,K,T)$ becomes an optimal control problem, namely one in which the objective of the disturbance is to ensure that $x(t) \notin R$, $\forall t \in [0,T]$. It then comes as little surprise that, under certain technical conditions, this set can be computed from the solution of an appropriate Hamilton-Jacobi-Bellman (HJB) equation (Bardi and Capuzzo-Dolcetta, 1997).

More specifically, we assume that the target set R is closed can be represented as the zero sublevel set of a bounded Lipschitz continuous function $\phi_R : \mathbb{R}^n \to \mathbb{R}$ such that

$$R = \{x \in \mathbb{R}^n : \phi_R(x) \leq 0\}.$$

The function ϕ_R is sometimes referred to as the level set representation of R (Sethian, 1999; Osher and Fedkiw, 2002). Now consider the HJB equation

$$\frac{\partial \phi}{\partial t} + \min\left[0, H\left(x, \frac{\partial \phi}{\partial x}\right)\right] = 0,\ \phi(x,0) = \phi_R(x) \tag{2.3}$$

with the optimal Hamiltonian

$$H(x,p) = \max_{d \in D} p^T f(x, K(x), d). \tag{2.4}$$

Let $\phi : \mathbb{R}^n \times [-T, 0] \to \mathbb{R}$ be the unique viscosity solution (Crandall and Lions, 1983) to (2.3) and (2.4). Then by a special case of the argument presented in Mitchell et al. (2005),

$$\mathscr{R}(R,K,T) = \{x \in \mathbb{R}^n : \phi(x,-T) \leq 0\}.$$

On a computational note, numerical solutions of H-J equations can be calculated on a grid of the continuous state space $\mathbb{R}^n$ using the MATLAB Toolbox for Level Set Methods developed by Mitchell (2007*a*). It is based upon an implementation of the level set theory and computational methodologies described extensively in the texts by Osher and Fedkiw (2002) and Sethian (1999). The numerical solutions provide convergent approximations of the true solutions of (2.3) as the grid size is refined. However, in order to obtain accurate approximations, the computational complexity scales exponentially in the number of continuous dimensions. This currently limits the application of this method to continuous models with $n \leq 5$.

2.5.2 Unsafe Set

For a given avoid set $A \subset \mathbb{R}^n$, and time horizon $T \geq 0$, the unsafe set of (2.2) is the set of initial conditions x_0 for which regardless of the choice of control strategy, there exists a choice of disturbance strategy and a time instant $t \in [0,T]$ such that $x(t) \in A$. If one were to view this as a zero-sum differential game in which the objective of the control is to avoid the set A over $[0,T]$, then this is the set of winning initial conditions for the disturbance. As before, we will consider a definition for this set in the case that the control strategy is fixed.

Definition 2.7 (Unsafe Set)**.** Given an avoid set A, a time horizon T, and a Lipschitz continuous feedback law $K : \mathbb{R}^n \to U$, the unsafe set $\mathscr{A}(A,K,T)$ of (2.2) is given by

$$\mathscr{A}(A,K,T) = \{x_0 \in X : \exists d(\cdot) \in \mathscr{D}_T, \exists t \in [0,T],\ x(t) \in A\},$$

where $x(\cdot)$ is the solution of $\dot{x}(t) = f(x(t), K(x(t)), d(t)),\ x(0) = x_0$ on the interval $[0,T]$.

From this definition, it can be observed that the only difference between a capture set and an unsafe set lies in the objective of the control. Namely, in the former case, the control tries to reach some terminal set R, while in the latter case, it tries to avoid some terminal set A. Correspondingly, computation of the unsafe set proceeds by minor modification of the method given for the capture

set. In particular, we assume as before that there exists a bounded Lipschitz continuous function $\phi_A : \mathbb{R}^n \to \mathbb{R}$ such that

$$A = \{x \in \mathbb{R}^n : \phi_A(x) \leq 0\}.$$

Consider the HJB equation as given in (2.3) with the optimal Hamiltonian

$$H(x,p) = \min_{d \in \mathbb{D}} p^T f(x, K(x), d). \tag{2.5}$$

Let $\phi : \mathbb{R}^n \times [-T, 0] \to \mathbb{R}$ be the unique viscosity solution to (2.3) and (2.5). Then by another application of the argument presented in Mitchell et al. (2005),

$$\mathscr{A}(A,K,T) = \{x \in \mathbb{R}^n : \phi(x,-T) \leq 0\}.$$

For the discussions on controller design, it is important to note that the complement of the unsafe set, denoted as $\mathscr{A}^C(A,K,T) := \mathbb{R}^n \setminus \mathscr{A}(A,K,T)$, is the set of initial conditions x_0 for which the trajectory of (2.2) under control law K avoids the set A over $[0,T]$, regardless of any admissible disturbance realization $d(\cdot) \in \mathscr{D}_T$.

2.5.3 Invariant Set

For a given set $W \subset \mathbb{R}^n$, an invariant subset of W under (2.2) is a set of initial conditions x_0 for which there exists a choice of control strategy such that, regardless of the choice of disturbance strategy, the trajectory of (2.2) satisfies $x(t) \in W$, $\forall t \geq 0$. The union of all such sets is called a maximal invariant set. A definition of this set is given below for the case in which the control strategy is fixed.

Definition 2.8 (Maximal Invariant Set)**.** Given a set $W \subset \mathbb{R}^n$ and a Lipschitz continuous feedback law $K : \mathbb{R}^n \to U$, then the maximal invariant set $Inv(W,K)$ of (2.2) is given by

$$Inv(W,K) = \{x_0 \in X : \forall d(\cdot) \in \mathscr{D}_T, \forall t \geq 0,\ x(t) \in W\},$$

where $x(\cdot)$ is the solution of $\dot{x}(t) = f(x(t), K(x(t)), d(t))$, $x(0) = x_0$ on the interval $[0,T]$.

As discussed in Tomlin et al. (2000), this set can be computed as an extension of the finite horizon unsafe set computation to the infinite horizon case. In particular, the problem can be viewed as a zero-sum differential game in which the objective of the control is to avoid the complement of W at all times.

Over any finite time horizon $[0,T]$, the set of winning initial conditions for the control in this differential game can be computed as $\mathscr{A}^C(\mathbb{R}^n \setminus W, K, T)$, using the procedures described in Section 2.5.2. Let $\phi_T : \mathbb{R}^n \to \mathbb{R}$ be a level set representation of this set. Then if ϕ_T converges to some function ϕ^* as $T \to \infty$, namely if the dynamic programming procedure as described by the HJB equation (2.3) and (2.5) converges to a fixed point, then ϕ^* provides a level set representation of the maximal invariant set

$$Inv(W,K) = \{x \in \mathbb{R}^n : \phi^*(x) \leq 0\}.$$

Furthermore, it can be verified that this set is invariant with respect to itself, namely for every initial condition $x_0 \in Inv(W,K)$, the trajectory of (2.2) under control law K satisfies $x(t) \in Inv(W,K)$, $\forall t \geq 0$, regardless of the disturbance realization.

2.6 Controller Design Procedures

As discussed in section 2.4.3, through an appropriate choice of switching policy as per equation (2.1), the task of controller design for problems 2.1 and 2.2 can be formulated in terms of a collection of continuous reachability problems. The continuous control design, however, needs to ensure both the reachability specification for each discrete state, as well as compatibility conditions between successive discrete states. In this section, we describe procedures for performing this control design, using reachability analysis as a design tool.

For notational convenience, the subscript q will be used to denote capture sets, unsafe sets, and invariant sets computed for a particular mode in the mode sequence. In particular, for a given target set $R \subset \mathbb{R}^n$, time horizon $T \geq 0$, and feedback law $K : \mathbb{R}^n \to U$, the capture set under the continuous dynamics in mode $q_i \in Q$ is denoted as $\mathscr{R}_{q_i}(R, K, T)$.

2.6.1 Automated Sequential Transition Systems with Reach-avoid Specifications

We first present a design procedure for Problem 2.1. In particular, during each phase of the design procedure, we design a control law for mode q_i to ensure that the target set R_i can be attained. Reachability calculations are then performed to check whether a compatibility condition is met, namely whether the set of initial conditions satisfying the reach-avoid objectives under this control law contains the target set of the previous discrete state q_{i-1}. The control law is then adjusted as necessary to satisfy this condition. This is described more precisely below.

Let $\mathscr{H}$ be an automated sequential transition system $\mathscr{H}$, such that the vector field f of $\mathscr{H}$ satisfies Assumption 2.2 for each discrete state $q \in Q$. Then given target sets $R_i \subset X$, $i = 1, ..., m$, and avoid sets $A_i \subset X$, $i = 1, ..., m$, a control policy (F, K) can be designed using the following procedure, starting with mode q_m.

1. Design a continuous control law $K(q_i, \cdot)$ which regulates initial conditions in R_{i-1} to the target set R_i, under the continuous dynamics $\dot{x} = f(q_i, x, K(q_i, x), d)$.

2. Compute the capture set under this control law to the first time instant τ_i, such that $R_{i-1} \subset \mathscr{R}_{q_i}(R_i, K(q_i, \cdot), \tau_i)$.

3. Compute over the time interval $[0, \tau_i]$ the corresponding unsafe set $\mathscr{A}_{q_i}(A_i, K(q_i, \cdot), \tau_i)$.

4. Check if the condition $R_{i-1} \subset \mathscr{A}^C_{q_i}(A_i, K(q_i, \cdot), \tau_i)$ holds. If this condition does not hold, return to step 1 to modify the design of $K(q_i, \cdot)$. Otherwise, the control design for mode q_i is complete.

5. Repeat steps 1-4 for q_{i-1} until q_1. For q_1, set $R_0 = X_0$.

6. Choose a switching policy F according to (2.1).

It can be verified using the conditions given in Lygeros et al. (1999*a*) that under the choice of control policy (F,K) as designed above, the automated sequential transition system $\mathscr{H}$ has a unique infinite execution. Furthermore, using the definition of capture sets and unsafe sets as given in section 2.5, it can be verified that this execution satisfies the specifications of Problem 2.1. In particular, by steps 1-4 of the design procedure, the feedback law K satisfies

$$R_{i-1} \subset \mathscr{R}_{q_i}(R_i, K(q_i,\cdot), \tau_i) \cap \mathscr{A}^C_{q_i}(A_i, K(q_i,\cdot), \tau_i), \tag{2.6}$$

for every $i = 1,...,m$, with $R_0 = X_0$. This ensures that, for each mode q_i, any continuous trajectory initialized from within R_{i-1} will reach R_i within τ_i time units while avoiding A_i, regardless of the realization of the disturbance d. Furthermore, given the choice of switching policy F, continuous state evolution in each mode q_i is assured to be only initialized from within R_{i-1}. The desired properties then follow.

Remark 2.1. The control design in step 1 can be viewed as a reference tracking problem, for which a number of design methods have been proposed in the nonlinear control literature (see for example Sastry, 1999). In particular, one can choose a point $\bar{x} \in R_i$ as a constant reference and design a controller in the relative coordinates $\tilde{x} := x - \bar{x}$. However, the difficulty lies in the need to satisfy a safety constraint on x, an input constraint on u, possibly in the presence of a disturbance d. In chapter 3, we discuss a reachability-based approach to this problem in terms of switching control policies.

Remark 2.2. The choice of compatibility condition (2.6) is somewhat conservative due to the fact there may exist initial conditions in the unsafe set $\mathscr{A}_{q_i}(A_i, K(q_i,\cdot), \tau_i)$ which reaches R_i before A_i, but is nonetheless precluded and conservatively labeled unsafe. To reduce this conservatism, a modified reachability calculation combining target attainability and safety objectives can be performed, by solving a *constrained* H-J PDE (Mitchell, 2002). This would then replace the capture sets and unsafe sets in the control design procedure. The method given here is chosen for simplicity of presentation and ease of computation.

2.6.2 Semi-Automated Sequential Transition Systems with Reach-avoid and Invariance Specifications

Next, we present a design procedure for Problem 2.2. In this case, the reach-avoid objectives for the transition states q_i can still be satisfied by following a similar procedure as described in the preceding section. However, some additional design steps are necessary in order to ensure that the invariance objectives are met, and that the stationary states are properly composed with the transition states. The precise sequence of steps is given below.

Let $\mathscr{H}$ be a semi-automated sequential transition system $\mathscr{H}$, such that the vector field f of $\mathscr{H}$ satisfies Assumption 2.2 for each discrete state $q \in Q$. Then given target sets $R_i \subset X$, $i = 1,...,m$, avoid sets $A_i \subset X$, $i = 1,...,m$, and target neighborhoods $W_i \subset X$ satisfying $R_i \subseteq W_i \subset A_i^C$, a control policy (F,K) can be designed using the following procedure, starting with mode $\hat{q}_m$.

1. Design a continuous control law $K(\hat{q}_i,\cdot)$ which ensures that trajectory initialized from within R_i (or a subset thereof) stays within W_i, under the dynamics $\dot{x} = f(\hat{q}_i,x,K(\hat{q}_i,x),d)$.

2. Compute the maximal invariant set $Inv(W_i,K(\hat{q}_i,\cdot))$ under this control law.

3. If $Inv(W_i,K(\hat{q}_i,\cdot)) \cap R_i \neq \emptyset$, choose a target set $\tilde{R}_i \subseteq Inv(W_i,K(\hat{q}_i,\cdot)) \cap R_i$. Otherwise, return to step 1 to modify the design of $K(\hat{q}_i,\cdot)$.

4. Design a continuous control law $K(q_i,\cdot)$ which regulates initial conditions in W_{i-1} to the target set $\tilde{R}_i$, under the continuous dynamics $\dot{x} = f(q_i,x,K(q_i,x),d)$.

5. Compute the capture set under this control law to the first time instant t_i, such that $W_{i-1} \subset \mathscr{R}_{q_i}(\tilde{R}_i,K(q_i,\cdot),\tau_i)$.

6. Compute over the time interval $[0,\tau_i]$ the corresponding unsafe set $\mathscr{A}_{q_i}(A_i,K(q_i,\cdot),\tau_i)$.

7. Check if the condition $W_{i-1} \subset \mathscr{A}^C_{q_i}(A_i,K(q_i,\cdot),\tau_i)$ holds. If this condition does not hold, return to step 4 to modify the design of $K(q_i,\cdot)$. Otherwise, the control design for mode q_i is complete.

8. Repeat steps 1-7 for q_{i-1} and $\hat{q}_{i-1}$ until q_1. For q_1, set $W_0 = X_0$.

9. Choose a switching policy F according to (2.1), but replacing the switching region R_i by $\tilde{R}_i$.

It can be verified using the conditions given in Lygeros et al. (1999*a*) that under the choice of control policy (F,K) as designed above, the semi-automated sequential transition system $\mathscr{H}$ has a unique infinite execution. Furthermore, using the definition of capture sets, unsafe sets, and invariant sets as given in section 2.5, it can be verified that this execution satisfies the specifications of Problem 2.2. In particular, by steps 1-3 of the design procedure, the feedback law K ensures that, for each mode $\hat{q}_i$, trajectories initialized from $\tilde{R}_i \subseteq R_i$ stays within W_i, for every $i = 1,...,m$ and admissible disturbance realization. Furthermore, by steps 4-7 of the design procedure, the feedback law K satisfies

$$W_{i-1} \subset \mathscr{R}_{q_i}(\tilde{R}_i,K(q_i,\cdot),\tau_i) \cap \mathscr{A}^C_{q_i}(A_i,K(q_i,\cdot),\tau_i), \tag{2.7}$$

for every $i = 1,...,m$, with $W_0 = X_0$. This ensures that, for each mode q_i, any continuous trajectory initialized from within W_{i-1} will reach $\tilde{R}_i \subseteq R_i$ within τ_i time units while avoiding A_i, regardless of the realization of the disturbance d. Given the choice of switching policy F, continuous state evolution in each mode $\hat{q}_i$ is assured to be only initialized from within $\tilde{R}_i$. On the other hand, due to the invariance property of K in the stationary modes, continuous state evolution in each mode q_i is assured to be only initialized from within W_{i-1}. The desired properties then follow.

Remark 2.3. For certain specifications of the target set R_i, target neighborhood W_i, continuous dynamics f, and input bounds U and D, there may not exist a subset $\tilde{R}_i$ of R_i such that every trajectory initiated from $\tilde{R}_i$ remains inside W_i at all times. To check the feasibility of an invariance

objective, one may consider performing an invariant set calculation using a differential game formulation of the problem as described in Tomlin et al. (2000), and verify the condition given in step 3. In the case that an invariance objective is found to be infeasible, one may consider modifying the specification of the target set R_i or the target neighborhood W_i.

Remark 2.4. The control design in step 1 can be viewed as a stabilization problem, by choosing some point $\bar{x} \in R_i$ and designing a stabilizing controller in the relative coordinate system $\tilde{x} := x - \bar{x}$. For nonlinear systems, this design can be performed for example using Lyapunov-based techniques (Sastry, 1999). However, the difficulty again lies in finding control designs which satisfies the state constraint W_i and the input constraint U, while accounting for continuous disturbances. In chapter 3, a reachability-based approach to this problem will be discussed.

2.7 Recovery from Improper Initialization

The design procedures described in the preceding section provides assurances that under operating conditions which satisfies the assumptions of the system model, the desired specifications will be achieved. However, given the myriad of contingency scenarios which can arise during actual system operation, a system designer also needs to account for run-time fault conditions which causes the assumptions of the system model to be violated. For certain classes of faults, appropriate design choices can be made to enable recovery from the fault condition in an automatic fashion, for example through built-in redundancies. Nonetheless, due to the fact that not every contingency scenario can be anticipated at design time, some level of human supervision may be inevitable in the event of a fault condition.

In this section, we will discuss a possible use for reachable sets as a visual aid to guide human decision-making in the case of improper system initialization. Specifically, this is a scenario in which the state of the sequential transition system is initialized outside of the designated set *Init*, namely $(q_0, x_0) \notin Init$. Within the context of the aerial refueling example, this corresponds to a scenario in which the refueling sequence is initiated from a position outside the first waypoint set X_0. Such a scenario could arise for example due to operator mistakes, miscommunication between the UAV operator and the tanker pilot, or complex missions with multiple aircraft operating in proximity of each other. Using reachable sets as visual guidance can be helpful for motion planning applications in which the sets are computed in the planning space and hence provides the operator with a sense of the reachable space of the underlying continuous system.

To address this fault condition, the system designer can consider adding a finite number of general purpose recovery modes $\{\tilde{q}_1, \tilde{q}_2, ..., \tilde{q}_M\}$, corresponding to dynamics $\dot{x} = f(\tilde{q}_i, x, u, d)$, with choices of feedback laws $K(\tilde{q}_i, \cdot)$. In the case of AAR, this could for example be a set of escape maneuvers. The problem of recovering from the fault condition then becomes one of constructing a recovery sequence from the library of recovery modes at run-time, in order to drive the continuous state of the system into the feasible set of a mode $q \in Q$ of the original sequential transition system. This feasible set can be for example derived from the compatibility condition given in (2.6).

First, for each recovery mode $\tilde{q}_i$, an unsafe set computation can be performed at design time to determine the set of unsafe initial conditions $\mathscr{A}_{\tilde{q}_i}(A, K(\tilde{q}_i, \cdot), \tilde{\tau}_i)$, over some appropriate choice of

time horizon $\tilde{\tau}_i$. The time horizon should be long enough so that the unsafe set does not provide misleading information to the human operator, but also not so long that the resulting decisions are rendered excessively conservative. Thus, appropriate choices of time horizons should be tailored to the particular application.

At run-time, a human operator can consult these sets to determine the choice of recovery modes that can be safely initiated. In particular, as long as the system is initialized at a state outside the intersection of unsafe sets over all recovery modes, namely $x_0 \notin \bigcap_{\tilde{q}_i} \mathscr{A}_{\tilde{q}_i}(A, K(\tilde{q}_i, \cdot), \tilde{\tau}_i)$, at least one safe recovery mode $\tilde{q}_i$ is available. From the set of safe recovery modes, a particular mode can be then selected so as to make progress towards the feasible set of a transition state or stationary state in the sequential transition system. During the execution of this maneuver over time interval $[0, \tilde{\tau}_i]$, the operator can consult the computed unsafe sets and plan the next recovery mode $\tilde{q}_j$ in the fault recovery sequence. At a time when it is safe to perform maneuver $\tilde{q}_j$, a command can be issued to transition, and the procedure repeats until the system state recovers in a mode of the sequential transition system (not necessarily q_1).

2.8 Aerial Refueling Example

This section describes an application of the controller design methodology developed in this chapter to the specific example of Automated Aerial Refueling (AAR). The discussion will primarily focus on the case in which AAR is to be performed autonomously without human supervision (Problem 2.1) in order to illustrate the basic mechanics of the design procedures introduced in section 2.6. However, later on in the section, we will also briefly touch on the extension to invariance objectives, the use of reachable sets for fault recovery, and the effects of disturbances.

2.8.1 Overview of Automated Aerial Refueling (AAR) Process

In an aerial refueling process, a formation of unmanned aerial vehicles (UAVs) approaches a human piloted tanker aircraft. One by one, the UAVs perform a sequence of maneuvers to dock with a human operated fuel boom and then return to formation. A graphical top down view of the refueling process is shown in Fig. 2.2.

The tanker aircraft is shown in the center, with the refueling UAV flying in formation to be refueled. In the actual refueling process, the UAV typically approaches from a fixed position in the formation. For modeling purposes, the aircraft to be refueled is assumed to approach from a position behind and to the right of the tanker aircraft. From this position, the UAV initiates a sequence of maneuvers through the numbered waypoints, under a combination of human operator commands and autonomous decisions. The sequence of maneuvers in the AAR process are shown in Table 2.1. The model described here utilize the separation of waypoints found in the work of Ross et al. (2006).

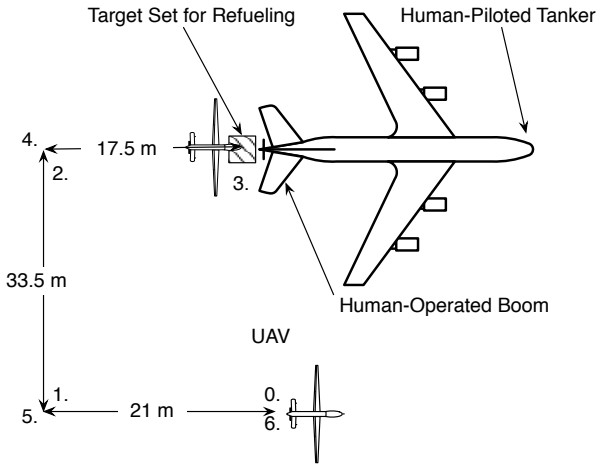


Figure 2.2: Diagram of waypoint locations in aerial refueling process, as labeled 1 through 6. Each flight maneuver corresponds to a transition between waypoints.

Event	**Maneuver**	**Man.#**	**Description**
σ_{12}	Detach 1	1	a single UAV detaches from a formation of UAVs in flight to a position slightly behind and to the right of a tanker aircraft.
σ_{23}	Precontact	2	the UAV banks left towards a position directly behind the tanker aircraft.
σ_{34}	Contact	3	the UAV approaches the tanker aircraft from behind to allow the boom operator on board the tanker to lower the fuel boom and catch the UAV.
σ_{45}	Postcontact	4	the UAV slows down and moves away from the tanker aircraft after the boom operator detaches the fuel boom.
σ_{56}	Detach 2	5	the UAV banks right towards a position directly behind the UAV formation.
σ_{67}	Rejoin	6	the UAV speeds up and rejoins the formation to complete the refuel sequence.

Table 2.1: Descriptions of flight maneuvers in the aerial refueling process.

2.8.2 Aircraft Model

Under the assumption that refueling occurs one vehicle at a time, we will focus our attention on the interaction between a single UAV and the tanker aircraft. The kinematics model as described here for the relative dynamics between the two aircraft leverages previous work Tomlin et al. (2001) in the modeling of aircraft conflict resolution scenarios in air traffic management. The model assumes that the two aircraft do not change altitude significantly in performing the aerial refueling maneuvers, and this is justified in the state of the practice for human-piloted maneuvers of this kind. In fact, using a change in altitude might jeopardize the success of the mission, as a boom operator might suspend the mission if loss of line of sight occurs; thus, there is motivation to preserve a 2D solution. Recent work by Williamson et al. (2009) provides promise that autonomous vehicles will be capable of sufficiently accurate onboard sensing to utilize the selected coordinate system. Placing the two aircraft in a 2D plane, the relative motion of the two aircraft in the UAV reference frame can be modeled as:

$$\dot{x} = f(x,u,d) = \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -u_1 + d_1\cos x_3 + u_2 x_2 \\ d_1 \sin x_3 - u_2 x_1 \\ -u_2 \end{bmatrix} \tag{2.8}$$

where x_1, x_2, x_3 are the longitudinal, lateral, and heading coordinates of the tanker aircraft in the UAV reference frame, u_1, u_2 are the translational and angular velocities of the UAV as indicated in Fig. 2.3, and d_1 is the translational velocity of the tanker aircraft. Here it is assumed that the tanker is in straight and level flight, and hence its angular velocity is set to zero. For most of our reachability and simulation results, it is assumed that the tanker aircraft maintains a nominal forward velocity v_0. However, as discussed in section 2.5, the reachability computation can be modified in a straightforward manner to account for fluctuations in the tanker velocity within a bounded range, and this case is covered in section 2.8.10, which demonstrates the corresponding changes to the reachable set calculations.

With regards to parameter values, the nominal velocity of the tanker aircraft is chosen to be $v_0 = 84.8$ m/s (75% of the maximum allowable velocity of the UAV); the velocity input u_1 for the UAV has the saturation limits $[40, 113]$ m/s, and the angular velocity input u_2 has the saturation limits $[-\pi/6, \pi/6]$ s^{-1}. The maximum UAV velocity value is based on published specifications for the MQ-9 Predator B; other values are chosen based on realistic constraints.

For completeness, it should be noted that the relative coordinates in the UAV reference frame and the tanker reference frame are related by a nonlinear coordinate transformation. Specifically, suppose $x = (x_1, x_2, x_3)$ is the coordinates of the tanker in the UAV reference frame, and $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)$ is the coordinates of the UAV in the tanker reference frame, then $\tilde{x} = \rho(x)$, where $\rho : \mathbb{R}^3 \to \mathbb{R}^3$ is given by

$$\rho\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = \begin{bmatrix} -x_1\cos x_3 - x_2 \sin x_3 \\ x_1 \sin x_3 - x_2 \cos x_3 \\ -x_3 \end{bmatrix} \tag{2.9}$$

This transformation will be used in transforming target sets and avoid sets specified in tanker coordinates into UAV coordinates. Specifically, suppose a set $\tilde{S}$ is represented by a function $\tilde{\phi}$ in the tanker reference frame (namely $\tilde{\phi}(\tilde{x}) \leq 0$, $\forall \tilde{x} \in \tilde{S}$), then the corresponding set S in the UAV reference frame is represented by the function $\phi = \tilde{\phi} \circ \rho$.

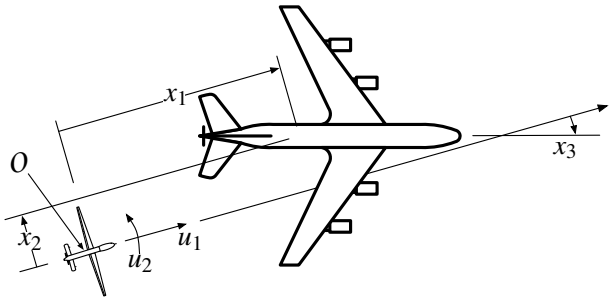


Figure 2.3: Relative-coordinate system, kinematic model. The origin of the coordinate system is centered on the UAV.

2.8.3 Hybrid System Abstraction of AAR

The sequence of flight maneuvers as described in Table 2.1, along with the kinematics model of aircraft dynamics as given in (2.8) provides us with an abstraction of aerial refueling process in terms of a sequential transition system, as shown in Fig. 2.4. In this model, the transition states consists of the sequence of flight maneuvers as listed in Table 2.1, while the stationary states consists of intermediate maneuvers in which the UAV is to wait in a neighborhood of each waypoint while waiting for operator command. In addition, there are four general purpose escape maneuvers, labeled $\tilde{q}_1$ to $\tilde{q}_4$ to handle the case of improper initialization as discussed in section 2.7. The continuous dynamics within each flight maneuver is identical and given by (2.8). The various maneuvers differ only by the choice of continuous control laws $K(q_i, \cdot)$, $K(\hat{q}_i, \cdot)$, and $K(\tilde{q}_i, \cdot)$, corresponding to transition maneuvers, stationary maneuvers, and escape maneuvers, respectively.

We first consider the case in which the flight maneuvers are to be performed autonomously. By choosing a neighborhood of states around each waypoint in Fig. 2.2 as a target set, and by choosing a protected zone around the tanker aircraft to be an avoid set, the task of designing AAR can be formulated as an instance of Problem 2.1. As described in section 2.4 and section 2.6, through an appropriate choice of switching policy, the design parameters become the continuous control laws within the respective flight maneuvers. In the following, we will discuss the specification of the target sets and avoid sets, as well as the form of the continuous control laws.

2.8.4 Specification of Target Sets and Avoid Sets

The target set R_i for each maneuver q_i is chosen to be a disc shaped neighborhood around each desired waypoint (see Fig. 2.5), with bounds on the relative heading error. This choice is consistent

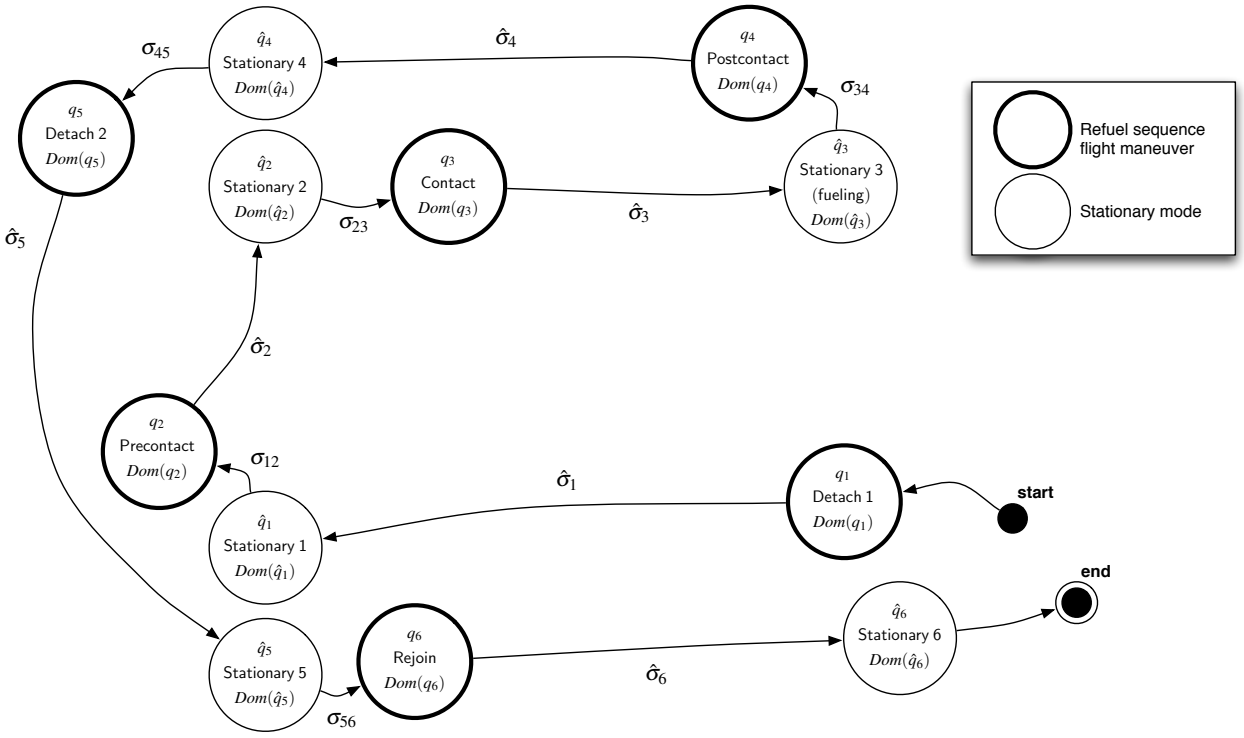


Figure 2.4: Discrete states and transitions in hybrid system abstraction of AAR process.

with the objective of controlling the aircraft to within some Euclidean distance of a given waypoint. For waypoint i, this set can be specified in tanker coordinates as $\tilde{R}_i = B([-x_{1f}(q_i),\ -x_{2f}(q_i)], r_0) \times [-\Delta\theta, \Delta\theta]$ where $B(x_0, r)$ denotes a ball of radius r, centered at x_0 in $\mathbb{R}^2$. In this case, the radius and heading tolerance are chosen to be $r_0 = 4$ m and $\Delta\theta = \pi/16$ rad, respectively. The corresponding set in the UAV coordinate frame is obtained from the transformation ρ in (2.9).

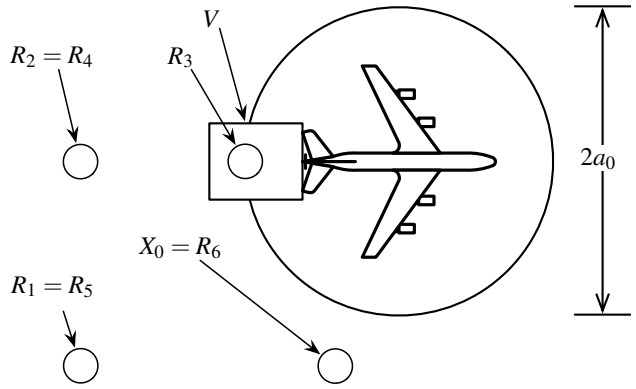


Figure 2.5: Target sets and avoid sets for transition maneuvers in AAR process.

Each flight maneuver uses an identical avoid set A, namely the set of continuous states corresponding to minimum separation infringement (MSI) violation between the tanker aircraft and UAV. This set consists of a disc in the x_1-x_2 plane, with a small neighborhood of states around the fuel boom removed to allow approach by the UAV. In the tanker reference frame, this is given by $\tilde{A} = (B([15,\ 0], a_0) \times [-\pi, \pi]) \setminus V, \forall q_i \in Q$, where $a_0 = 30\,\text{m}$ is the protected radius (chosen based upon published data of the wingspan of a Boeing KC-135 Stratotanker), the origin of the tanker's coordinate system is 15 m from the centroid of the tanker, and V is a small hyper-rectangle of states around the boom location, defined in the tanker reference frame as $V = \left\{\tilde{x} \in \mathbb{R}^3 : -15\text{m} \leq \tilde{x}_1 \leq 10\text{m}, -8\text{m} \leq \tilde{x}_2 \leq 8\text{m}, -\pi \leq \tilde{x}_3 \leq \pi\right\}$. The corresponding avoid set A in the UAV coordinate frame can be obtained from the coordinate transformation ρ.

2.8.5 Structure of Continuous Controllers

The feedback control laws to perform the various maneuvers are applied through the inputs u_1 and u_2. To emulate high-level waypoint following algorithms, proportional control laws are used to steer the UAV to the various desired waypoints. For transition maneuvers q_1 to q_6, the feedback laws $K(q_i, \cdot)$ are of the form

$$u_1 = k_1(x_1 - x_{1f}) + v_0 \tag{2.10}$$

$$u_2 = k_2(x_2 - x_{2f}) \tag{2.11}$$

where k_1 and k_2 are proportional gain constants, and x_{1f}, x_{2f} are the desired waypoint locations in the UAV reference frame. To take into account actuator limitations, the control laws are saturated to within the input ranges given in section 2.8.2. The control law $K(\hat{q}_i, \cdot)$ for each stationary maneuver $\hat{q}_i$, $i = 1, ..., 6$ is chosen to be identical as that of the preceding transition maneuver q_i.

The waypoint locations for the transition and stationary maneuvers are specified in Table 2.8.5. During the control design procedure, the proportional gain constants will be selected so as to ensure the reachability objective of each flight maneuver.

Maneuver	**Mode Label**	x_{1f}	x_{2f}
Detach 1, Stationary 1	$q_1, \hat{q}_1$	25.5	33.5
Precontact, Stationary 2	$q_2, \hat{q}_2$	25.5	0
Contact, Stationary 3	$q_3, \hat{q}_3$	8.0	0
Postcontact, Stationary 4	$q_4, \hat{q}_4$	25.5	0
Detach 2, Stationary 5	$q_5, \hat{q}_5$	25.5	33.5
Rejoin, Stationary 6	$q_6, \hat{q}_6$	4.5	33.5

Table 2.2: Desired waypoint locations for continuous control laws (x_{1f}, x_{2f}, in meters).

Finally, the control laws $K(\tilde{q}_i, \cdot)$ for the four escape maneuvers $\tilde{q}_i$, $i = 1, ..., 4$ are chosen as follows:

1. *Escape 1* (steer left at max speed): $u_1 = u_{1_{\max}}$, $u_2 = u_{2_{\max}}$;
2. *Escape 2* (steer right at max speed): $u_1 = u_{1_{\max}}$, $u_2 = -u_{2_{\max}}$;
3. *Escape 3* (slow down): $u_1 = u_{1_{\min}}$, $u_2 = 0$;
4. *Escape 4* (speed up): $u_1 = u_{1_{\max}}$, $u_2 = 0$.

2.8.6 Control Design Using Capture Sets and Collision Sets

We now describe the design the maneuver control laws, using the procedures given in section 2.6.1. For the rest of our discussions in this section, unsafe sets will be referred to interchangeably as *collision sets*, due to the fact that these sets correspond to initial conditions that could result in a collision with the tanker aircraft.

For a given flight maneuver q_i, we first fix a set of control gains and compute the capture set with respect to a target set R_i until a time instant τ_i such that $R_{i-1} \subset \mathscr{R}_{q_i}(R_i, K(q_i,\cdot), \tau_i)$. An unsafe set computation is then performed to check the safety condition $R_{i-1} \subset \mathscr{A}^C_{q_i}(A, K(q_i,\cdot), \tau_i)$. For mode q_1, the set R_0 is specified to be the set of permissible initial states X_0, as shown in Fig. 2.5. The control gains for each flight maneuver are then adjusted as necessary to ensure the target attainability and safety objectives are met. The set of control gains and maneuver timings obtained from this design procedure is summarized in Table 2.8.6.

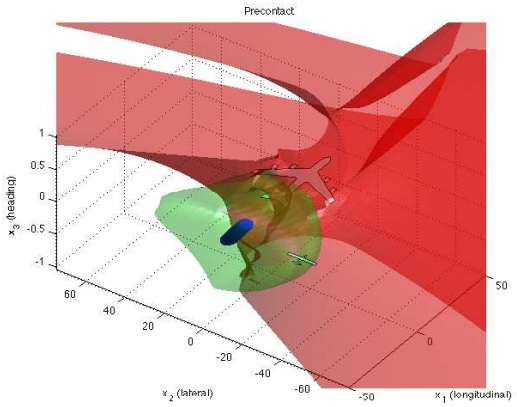
Maneuver	k_1	k_2	**Time τ_i (s)**	**Elapsed time (s)**
Detach 1	3	1	1.25	1.25
Precontact	0.5	5	3.00	4.25
Contact	2.5	1	1.00	5.25
Postcontact	2.5	1	1.00	6.25
Detach 2	1	5	3.50	9.75
Rejoin	3	1	1.25	11.0

Table 2.3: Proportional gain constants (k_1, k_2) and timings (τ_i in seconds) for transition maneuvers.

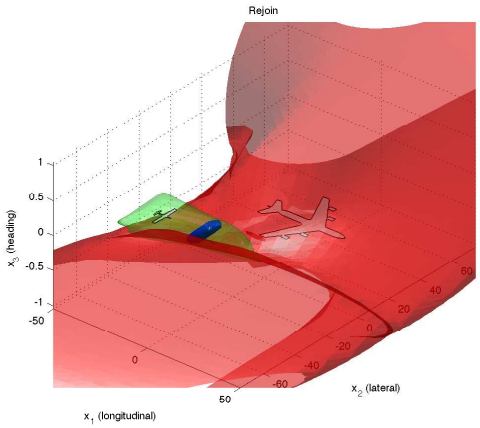
Example capture sets are shown in Fig. 2.6a and Fig. 2.6b for the Contact (q_3) and Rejoin (q_6) maneuvers. It can be seen that the control laws for the two maneuvers are designed so as to ensure that the target set of the preceding maneuver is contained within the capture set of the current maneuver and has empty intersection with the collision set of the current maneuver, thus ensuring proper composition between continuous trajectories of successive flight maneuvers in the refueling sequence.

2.8.7 Refueling Sequence Simulation

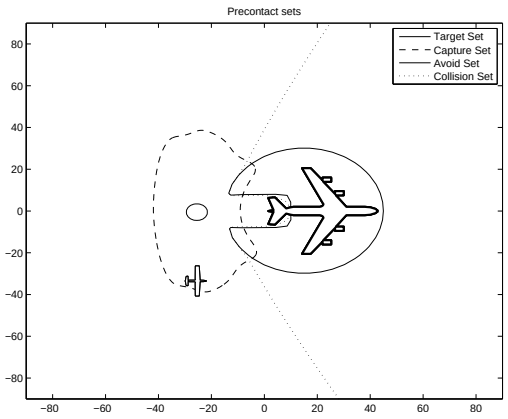
A complete simulation of the refueling sequence is constructed to check the satisfaction of the safety and target attainability objectives. In this simulation, the UAV does not spend any time in



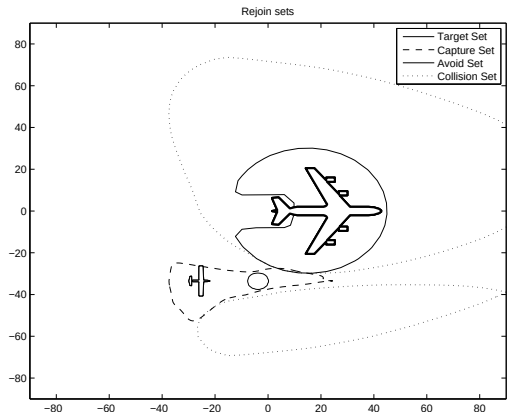
(a) Capture (light, green) and collision (dark, red) sets for Precontact.



(b) Capture (light, green) and collision (dark, red) sets for Rejoin.



(c) Slice of Precontact capture (dashed) and collision (dotted) sets at $x_3 = 0$



(d) Slice of Rejoin capture (dashed) and collision (dotted) sets at $x_3 = 0$

Figure 2.6: Capture sets and collision sets for Precontact and Rejoin maneuvers. In each figure, x_1 and x_2 represent longitudinal and lateral offset (respectively), and x_3 represents the offset in heading between the UAV and tanker.

the stationary modes, namely a forced transition is taken to the next maneuver in the sequence as the state of the UAV enters a target set.

Some snapshots of the simulation are shown in Fig. 2.7, where the capture sets and collision sets for each flight maneuver are superimposed on the trajectory of the UAV. As guaranteed by the mode switching conditions, each maneuver is completed within the transition timing given in Table 2.8.6, without entering the avoid set A corresponding to MSI. Furthermore, it is verified that whenever the system state x enters a target set R_i in mode q_i, the conditions $x \in \mathscr{R}_{q_{i+1}}(R_{i+1}, K(q_{i+1},\cdot), \tau_{i+1})$ and $x \notin \mathscr{A}_{q_{i+1}}(A, K(q_{i+1},\cdot), \tau_{i+1})$ are satisfied, thus ensuring the feasibility of the next flight maneuver.

2.8.8 Extension to Invariance Objectives

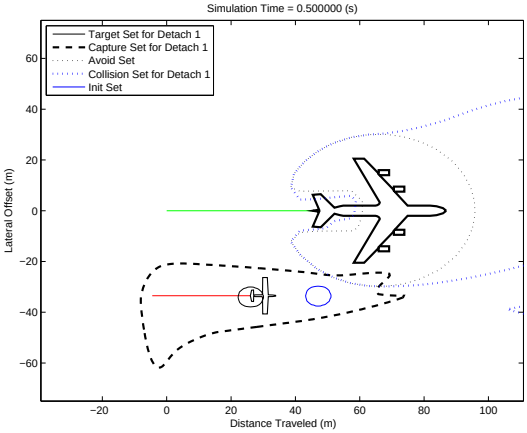
Here we consider an extension of the controller design to the case in which the specifications requires the UAV to remain in a neighborhood of certain waypoints while waiting for operator commands to proceed to the next flight maneuver. This falls within the framework of a semi-automonomous sequential transition system. In particular, we will focus on the controller design for Stationary 3 ($\hat{q}_3$), corresponding to when the UAV is expected to be refueling. In this case, it is necessary that the UAV maintains itself within a neighborhood of the fueling boom while the boom operator performs the refueling operation.

We specify a target neighborhood for this stationary maneuver in tanker coordinates as $W_3 = B([-x_{1f}(q_3),\ -x_{2f}(q_3)], r_1) \times [-\Delta\theta, \Delta\theta]$, where the waypoint location (x_{1f}, x_{2f}) is as given in Table 2.8.5 for the Contact maneuver, the neighborhood radius is set to $r_1 = 6$ m, and the heading tolerance is set to $\Delta\theta = \pi/16$ rad. The controller for this maneuver is chosen to be the same as that designed for the contact maneuver and an invariant set calculation is performed according to the procedure described in section 2.5.3. The result is shown in Fig. 2.8. In these plots, the invariant set satisfies $Inv(W_3, K(\hat{q}_3,\cdot)) \subset \mathscr{R}_{q_4}(R_4, K(q_4,\cdot), \tau_4)$ and $Inv(W_3, K(\hat{q}_3,\cdot)) \cap \mathscr{A}_{q_4}(A, K(q_4,\cdot), \tau_4) = \emptyset$, namely it lies within the feasible set of the next maneuver Postcontact (q_4) in the refueling sequence. In order to ensure that the contact maneuver ends in a state satisfying the invariance objective, the target set for the contact maneuver can be chosen according to the procedures of section 2.6.2 as $\tilde{R}_3 = B([-x_{1f}(q_3),\ -x_{2f}(q_3)], r_0) \times [-\pi/18, \pi/18]$, where r_0 is as given in section 2.8.4.

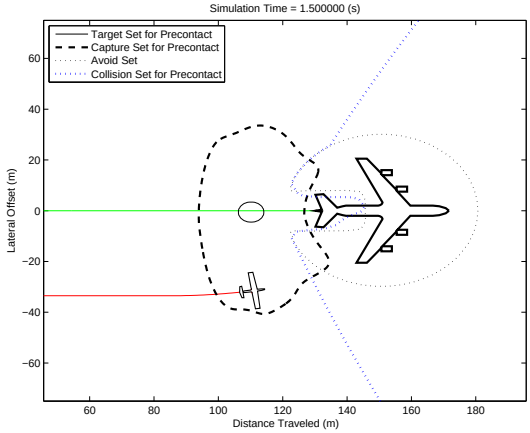
2.8.9 Scenario of Improper Initialization

In this section, we formulate a simulation scenario in which the system state is initialized outside the set X_0. This provides an example of improper initialization as discussed in section 2.6.2 and will be used to illustrate the use of reachable sets as a tool for guiding human decision making.

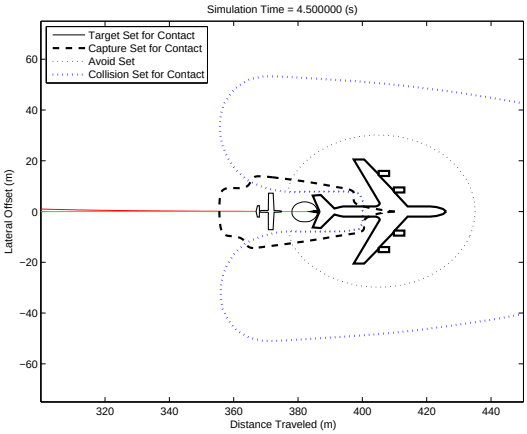
The goal in this case is to construct a sequence of escape maneuvers to arrive at the target set R_2 of the *Precontact* (q_2) maneuver, using the collision sets $\mathscr{A}_{\tilde{f}_i}(A, \tilde{K}_i, \tilde{\tau}_i)$ computed for escape maneuvers 1-4, as well as the capture and collision sets for the *Precontact* maneuver. In practice, this task would be carried out by a trained UAV operator. For this simulation scenario, however, the maneuver selection is performed by heuristic examination of the generated sets. The results are shown in Fig. 2.9.



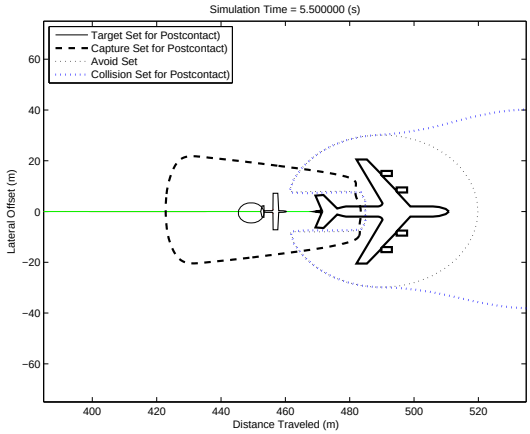
(a) Detach 1



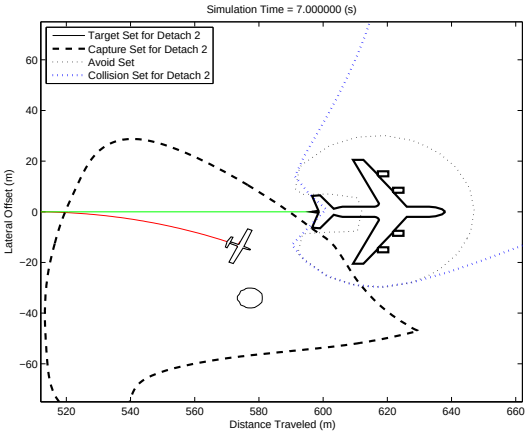
(b) Precontact



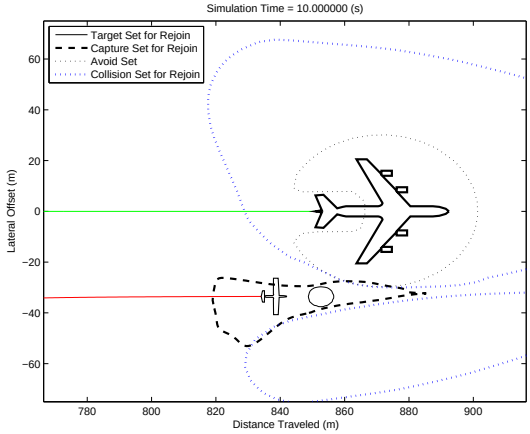
(c) Contact



(d) Postcontact

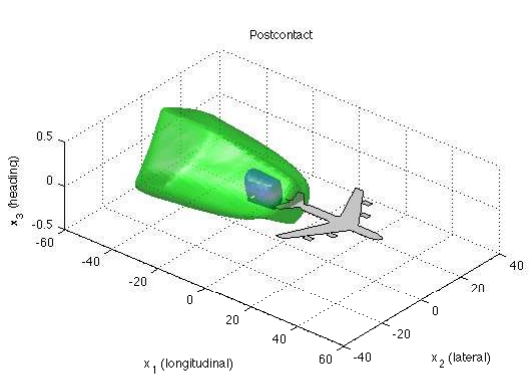


(e) Detach 2

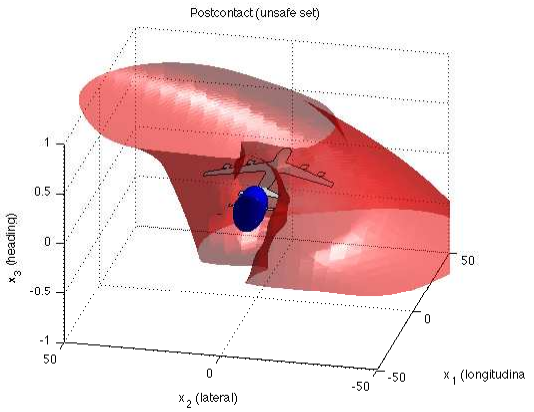


(f) Rejoin

Figure 2.7: Refueling sequence simulation with capture sets (dashed lines), avoid and collision sets (dotted lines).



(a) Postcontact capture set (light, green) and Stationary 3 invariant set (dark, blue).



(b) Postcontact collision set (light, red) and Stationary 3 invariant set (dark, blue).

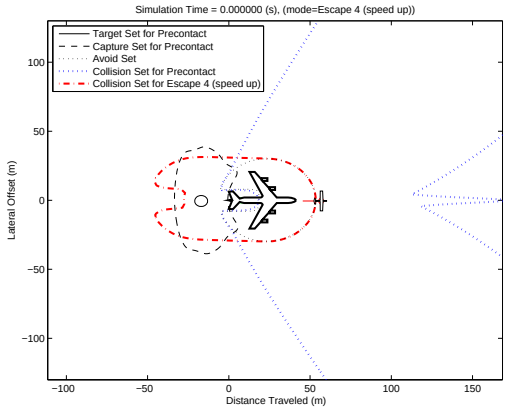
Figure 2.8: Results of an invariant set calculation for Stationary 3 maneuver, showing that the Postcontact maneuver can be safely initiated following refueling.

From the first plot, the UAV is initialized at a location outside the capture set of the Precontact maneuver ($x_0 \notin \mathscr{R}_{q_2}(R_2, K(q_2,\cdot), \tau_2)$). In fact, this initial condition lies inside the collision set of the Precontact maneuver ($x_0 \in \mathscr{A}_{q_2}(A, K(q_2,\cdot), \tau_2)$). In examining the collision sets computed for the escape maneuvers, it is found that $x_0 \notin \mathscr{A}_{\tilde{q}_4}(A, K(\tilde{q}_4,\cdot), \tilde{\tau}_4)$. The recovery maneuver Escape 4 ($\tilde{q}_4$) corresponding to "speed up" is then selected as a safe flight maneuver. After performing this maneuver for some time, while consulting the collision sets, it is found that both Escape 2 ($\tilde{q}_2$) and Escape 1 ($\tilde{q}_1$) become available, corresponding to "turn right" and "turn left", respectively. Maneuver Escape 2 is chosen first, followed by maneuver Escape 1 to return the heading to that of the tanker vehicle. Finally, maneuver Escape 3 is selected, corresponding to "slow down." While reducing speed, the state of the UAV enters the capture set of the *Precontact* maneuver ($x \in \mathscr{R}_{f_2}(R_2, K_2, \tau_2)$), and the UAV mode transitions to the *Precontact* maneuver, and the fault recovery sequence completes.

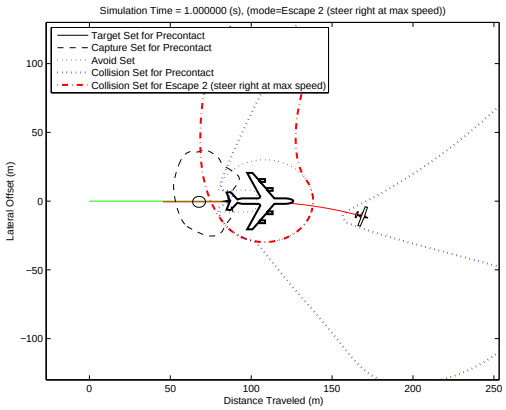
2.8.10 Effects of Disturbance on Reachable Set Computation

In the previous results, capture sets and collision sets are generated assuming a nominal tanker velocity of $v_0 = 84.8 m/s$. However, during execution time, there may be some degree of uncertainty associated with the velocity of the tanker, due to unmodeled dynamics and various environment disturbances (for example wind effects). This uncertainty may not be significant for maneuvers far enough from the tanker aircraft. However, for the *Contact* maneuver in which the UAV needs to come within close proximity of the tanker aircraft, even slight variations in the tanker aircraft speeds may compromise the safety of the maneuver.

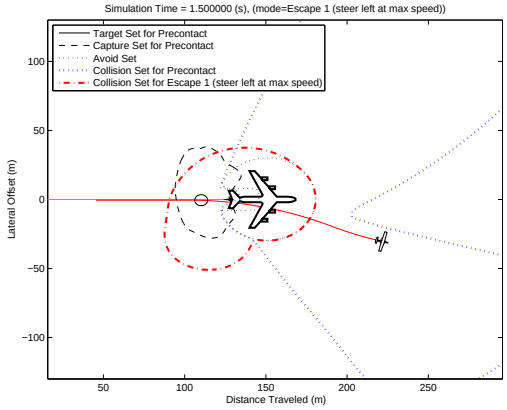
As discussed in section 2.5, the Hamilton-Jacobi method for reachability analysis offers the flexibility to account for this uncertainty in the tanker aircraft velocity. In this case, the tanker



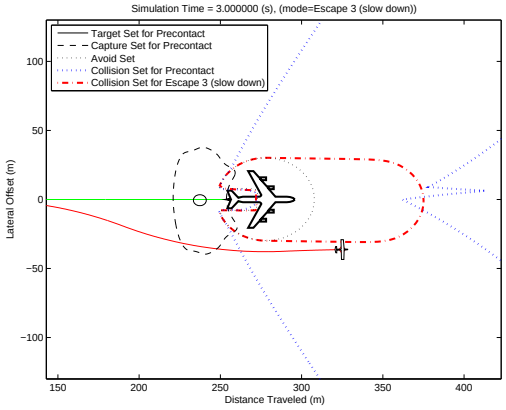
(a) Escape Mode 4 (Speed Up) initiated at $t = 0s$



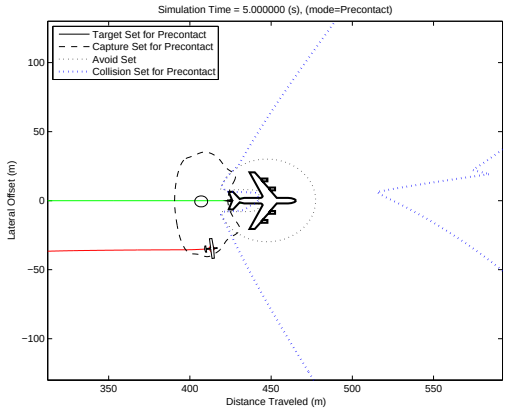
(b) Escape Mode 2 (Steer Right at Max Speed) was initiated at $t = 0.5s$, shown here at $t = 1.0s$



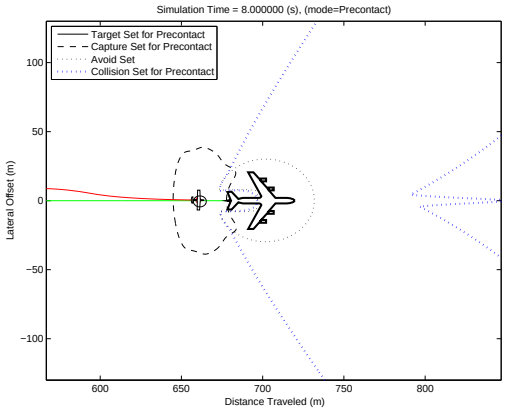
(c) Escape Mode 1 (Steer Left at Max Speed) initiated just before $t = 1.25s$



(d) Escape Mode 3 (Slow down), shown at $t = 3s$



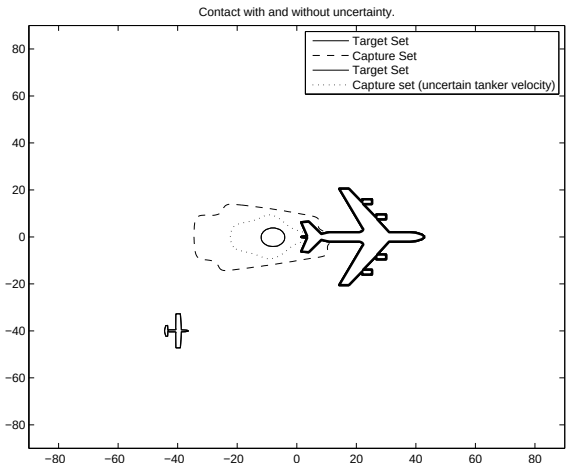
(e) Performing Precontact, shown here at $t = 5s$



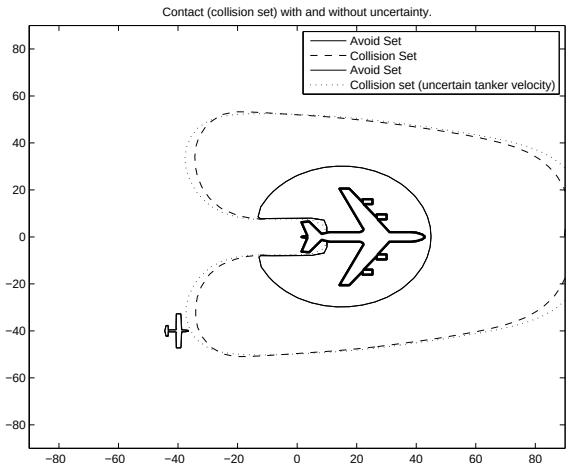
(f) Precontact completed, shown here at $t = 8s$

Figure 2.9: Fault recovery sequence simulation with capture set for Precontact (dashed lines), and collision sets for Precontact (dotted lines) and escape maneuvers (dash-dotted lines).

velocity d_1 is allowed to fluctuate in the bounded range $[79.14, 90.45]$ m/s (70-80% of the maximum allowable velocity of the UAV). The capture set and collision set for the Contact maneuver under the effects of this disturbance are shown in Fig. 2.10 (a) and (b), along with the same sets calculated under the nominal tanker velocity.



(a) Contact maneuver capture set without uncertainty (outer line), with uncertainty (inner line).



(b) Contact maneuver collision set with uncertainty (outer line), collision set without uncertainty (middle line), and the MSI (innermost line).

Figure 2.10: Capture set and collision set for Contact maneuver under worst-case tanker speed.

As expected, the capture set with added uncertainty is smaller than that without uncertainty, shown in Fig. 2.10a. This is due to the fact that under worst case tanker aircraft speed input, the tanker is effectively trying to prevent the UAV from entering the refueling zone. Similarly, the worst case collision set under uncertainty, shown in Fig. 2.10b, is larger than that without uncertainty. This results from the worst case tanker speed input which effectively tries to force a collision with the UAV.

Chapter 3

Controller Synthesis Algorithms for Safety and Reach-avoid Problems

3.1 Overview and Related Work

This chapter presents several computational algorithms for the synthesis of feedback control policies to satisfy safety and reach-avoid objectives for switched nonlinear systems. In particular, we consider an abstraction for high level control of a physical system as described in terms of a discrete decision process selecting amongst a finite set of continuous behaviors (e.g. maneuvers of an aircraft, motions of a robot, gears of an automobile), where the continuous behaviors are characterized by a nonlinear vector field, up to some bounded continuous disturbances. Using this abstraction, controller synthesis techniques are formulated to satisfy two types of specifications:

- *safety*: keep the system state within a prescribed safe set in the hybrid state space over finite or infinite time horizon;
- *reach-avoid*: drive the system state into a prescribed target set in the hybrid state space within finite time, subject to a constraint that the state trajectory avoids an unsafe set.

A control policy resulting from the controller synthesis algorithms consists of a set of feasible initial conditions, as well as a set-valued feedback law defined on the feasible set.

It is important to note that the switched system model employed in this chapter has a very different interpretation from the switched system model of the preceding chapter. Whereas the discrete states of a sequential transition system is used to represent *the phases of a dynamic process*, the discrete states here are used to represent *the set of continuous behaviors* that a high level controller can select from at any given time. In other words, the model of the preceding chapter represents a discretization or aggregation over the temporal space, while the model of the current chapter represents a discretization or aggregation over the control space. From this perspective, one can view the controller synthesis algorithms described here as a method for performing control design in each phase of the sequential reachability specification.

Safety and reach-avoid problems for switched nonlinear systems as considered in this chapter are special cases of hybrid reachability problems. As alluded to in the introduction and the preceding chapter, numerous theoretical and computational tools have been developed to address such problems over the past two decades, under varying assumptions on the hybrid system dynamics. Early efforts focused on timed automata and linear hybrid automata (Alur and Dill, 1994; Maler et al., 1995; Henzinger et al., 1997; Larsen et al., 1997; Yovine, 1997; Henzinger et al., 1998; Alur et al., 2000), in which case the simplicity of continuous dynamics allows an exact discrete abstraction of the hybrid system and the adaptation of discrete synthesis techniques to hybrid reachability problems. Although exact solutions to these problems are still possible for certain classes of hybrid systems with linear and nonlinear dynamics (Lafferriere et al., 1999; Shakernia et al., 2001; Del Vecchio, 2009), the consideration of general forms of continuous dynamics often requires approximation techniques. This has led to considerable research into methods for computing approximate continuous reachable sets.

One class of methods propagates explicit set representations such as polyhedra (Asarin et al., 2000*a*; Bemporad et al., 2000*b*; Chutinan and Krogh, 2003; Hwang et al., 2005; Han and Krogh, 2006), ellipsoids (Kurzhanski and Varaiya, 2000; Botchkarev and Tripakis, 2000), or zonotopes (Girard, 2005; Girard and Le Guernic, 2008) directly under flows of the system. Methods in this class typically consider linear systems or feedback linearizable nonlinear systems. Another class of methods approximates sets using representations defined on a discretized grid of the continuous state space, including approaches based upon viability theory (Cardaliaguet et al., 1999; Aubin et al., 2002; Saint-Pierre, 2002; Gao et al., 2007) and viscosity solutions of Hamilton-Jacobi equations (Mitchell et al., 2005; Mitchell, 2011). These approaches tend to be more general in the types of reachability computation and system dynamics that can be handled, but are often more computationally intensive.

Parallel to these efforts, which compute sets directly in the continuous state space, techniques have been proposed for computing approximate discrete abstractions of hybrid systems which allows the application of existing methods in discrete verification and supervisory control (Tiwari and Khanna, 2002, 2004; Kloetzer and Belta, 2006; Tabuada, 2008; Girard et al., 2010). Although computationally efficient for classes of systems with polynomial or affine dynamics, current instantiations of these techniques feature a similar type of growth in complexity as the viability and Hamilton-Jacobi approaches when general nonlinear systems are considered. Finally, for purposes of verifying safety, a computational technique based upon Lyapunov type analysis has also been proposed in Prajna et al. (2007) for systems with polynomial dynamics.

The reachable set computation and controller synthesis techniques described in this chapter are based upon the game theoretic framework for hybrid controller design as outlined in Lygeros et al. (1999*b*) and Tomlin et al. (2000), which formulates hybrid reachability problems as zero-sum dynamic games between the control and rational disturbances. The strength of this framework lies in its consideration of general forms of nonlinear continuous dynamics, as well as dynamic uncertainty modeled by bounded disturbance terms. The development of level set techniques for computing approximate solutions to Hamilton-Jacobi-Isaacs (HJI) equations (Mitchell et al., 2005) also promises accurate numerical computation of reachable sets, although restricted to systems of up to five continuous state dimensions due to a grid-based approximation. However, with the diffi-

culties inherited from analyzing nonlinear systems, continuous feedback policies in general cannot be derived in closed form. Furthermore, due to the interdependence of discrete and continuous dynamics, the problem of designing discrete controls also does not yield readily to automated synthesis algorithms. Thus, applications of this framework to practical problems such as automated highway platooning (Lygeros et al., 1998), flight envelope protection (Bayen et al., 2007), and aircraft conflict resolution (Tomlin et al., 2001) are often performed on a case by case basis with considerable insight from the control designer.

There has been an ongoing effort to develop computational algorithms for the synthesis of decision policies using Hamilton-Jacobi reachable sets. In Teo and Tomlin (2003), the authors describe a method for selecting evasive maneuvers for closely-spaced runway approaches, by checking the aircraft state information against unsafe sets computed for the evasive maneuvers. Due to the needs for fast online computation of reachable sets, the computation method is tailored to the particular application and assumes an open-loop selection of inputs. Another method is discussed in Hwang et al. (2005) for extracting control inputs from polytopic approximations of reachable sets for certain classes of nonlinear systems. An optimal selection of input, however, can be only determined along the boundaries of the approximating polytope, and may lead to chattering effects. The work described in Oishi et al. (2006) proposes an approach for selecting feedback linearizing control laws to achieve stabilization under safety constraints, based upon the results of a reachability calculation. However, issues of implementation and guarantees of safety and target attainability in applications with sampled state measurements and piecewise constant controls were not addressed.

The main contributions of our proposed methodology are as follows. First, we provide systematic procedures for the numerical computation of feedback control policies to satisfy hybrid reachability specifications for switched nonlinear systems. In particular, several reachability algorithms are proposed such that the output of each algorithm include both a set of initial conditions on which a given reachability objective is feasible, as well as a set-valued control law represented in terms of a collection of reachable sets. Second, we carry out analysis and synthesis tasks within the framework of a sampled-data system model. This ensures that the controllers computed through our algorithms will preserve the desired reachability specifications in *continuous time* even as state measurements and applications of control actions may be constrained to take place at *sampling instants*. Finally, we give detailed algorithms for the online selection of control inputs using the results of the offline reachability computations. These algorithms represent possible approaches to practically implement reachability-based controllers, by storing numerical representations of reachable sets and accessing them in an online setting as lookup tables.

The organization of this chapter is as follows. In section 3.2, we give a formal description of the sampled-data switched system model. In section 2.4, we formulate the safety and reach-avoid control problems within the context of this switched system model. In section 3.4, we provide a controller synthesis algorithm for the safety control problem, along with a numerical example of aircraft collision avoidance. In section 3.5, we propose a solution for the finite horizon reach-avoid problem, and illustrate the methodology through an experimental application on a quadrotor platform in section 3.6. Finally, we revisit the AAR example in section 3.7, in order to discuss the application of the proposed computational algorithms to switching controller design in sequential reachability problems.

3.2 Sampled-Data Switched System Model

We model a sampled-data switched system as a special case of the hybrid automaton discussed in section 2.3.1.

Definition 3.1 (Sampled-Data Switched System)**.** A sampled-data switched system is a tuple $\mathscr{H}_{sw} = (Q, X, \Sigma, U, D, \delta, f, T)$, defined as follows.

- *Discrete state space* $Q := \{q_1, q_2, ..., q_m\}$, $m \in \mathbb{N}$.
- *Continuous state space* $X := \mathbb{R}^n$.
- *Discrete input space* $\Sigma := \{\sigma_1, \sigma_2, ..., \sigma_{n_\sigma}\}$.
- *Continuous input space* U, a compact subset of $\mathbb{R}^{n_u}$.
- *Disturbance input space* D, a compact subset of $\mathbb{R}^{n_d}$.
- *Discrete transition function* $\delta : Q \times \Sigma \to Q$, describing the discrete state evolution.
- *Vector field* $f : Q \times X \times U \times D \to \mathbb{R}^n$, describing the continuous state evolution. It is assumed that f is uniformly continuous and bounded, and that for fixed $q \in Q$, $u \in U$, and $d \in D$, the function $x \to f(q, x, u, d)$ is Lipschitz continuous.
- *Sampling interval* $T > 0$.

As noted previously, the discrete states of this model have a different interpretation from the discrete states of the sequential transition system models given in sections 2.3.2 and 2.3.3. Specifically, the discrete state space Q of $\mathscr{H}_{sw}$ can be viewed as a set of operation modes that are provided as control choices to a high level controller, while the discrete state space of a sequential transition system can be viewed as a set of temporal phases in a dynamic process.

Informally, the executions of a sampled-data switched system proceeds as follows. At each sampling instant kT, we receive measurements of the system state $(q(kT), x(kT))$, and select based upon this information a discrete input $\sigma(kT) \in \Sigma$ and a continuous input $u(kT) \in U$, which are held constant on the sampling interval $[kT, (k+1)T)$. In response, the disturbance is allowed to select a realization $d : [kT, (k+1)T) \to D$. Given the switching command $\sigma(kT)$, the discrete state transitions to $\delta(q(kT), \sigma(kT)) \in Q$. The continuous state then evolves according to the vector field in the updated discrete state:

$$\dot{x}(t) = f(\delta(q(kT), \sigma(kT)), x(t), u(kT), d(t)), \tag{3.1}$$

for $t \in [kT, (k+1)T]$. Under the assumptions placed upon the vector field f, the existence and uniqueness of solutions to (3.1) is assured on each sampling interval. At the next time step, the discrete state is then given by $q((k+1)T) = \delta(q(kT), \sigma(kT))$, while the continuous state is given by $x((k+1)T)$ as obtained from the solution to (3.1), and the same process repeats.

More precisely, we allow control inputs to be chosen according to a set-valued feedback law defined as follows.

Definition 3.2 (Control Policy)**.** A control policy for $\mathscr{H}_{sw}$ is a sequence $\mu = (\mu_0, \mu_1, ...)$ of maps $\mu_k : Q \times X \to 2^{\Sigma \times U} \setminus \emptyset$. We denote the set of such admissible control policies by $\mathscr{M}$.

In particular, the feedback map μ_k provides a set of possible control inputs given a sampled state measurement $(q(kT), x(kT))$.

Under the worst-case assumption that the disturbance may be a rational adversary, we model a disturbance strategy for $\mathscr{H}_{sw}$ as a sequence of maps from the state and control input space to the set of admissible disturbance realizations. More specifically, consider the set of functions

$$\mathscr{D}_T = \{d : [0,T] \to D | d(\cdot) \text{ is measurable}\}.$$

Definition 3.3 (Disturbance Strategy)**.** A disturbance strategy for $\mathscr{H}_{sw}$ is a sequence $\gamma = (\gamma_0, \gamma_1, ...)$ of maps $\gamma_k : Q \times X \times \Sigma \times U \to \mathscr{D}_T$. We denote the set of such admissible disturbance strategies by Γ.

We can now give a formal definition for the executions of a sampled-data switched system under fixed choices of control policy and disturbance strategy.

Definition 3.4 (Switched System Execution)**.** For a given initial condition $(q_0, x_0) \in Q \times X$, control policy $\mu \in \mathscr{M}$, disturbance strategy $\gamma \in \Gamma$, and time horizon $N > 0$, the execution of a sampled-data switched system $\mathscr{H}_{sw}$ on $[0, NT]$ is a function $(q, x) : [0, NT] \to Q \times X$ as returned by the following algorithm.

Algorithm 3.2.1 Switched System Execution

Require: Initial condition $(q_0, x_0) \in Q \times X$, control policy $\mu \in \mathscr{M}$, and disturbance strategy $\gamma \in \Gamma$.

Set $q(0) \Leftarrow q_0$, $x(0) \Leftarrow x_0$;
for $k = 0$ to N **do**
  Choose $(\sigma(kT), u(kT)) \in \mu_k(q(kT), x(kT))$;
  Set $d = \gamma_k(q(kT), x(kT), \sigma(kT), u(kT))$;
  Set $q(t) = \delta(q(kT), \sigma(kT))$ for $t \in (kT, (k+1)T]$;
  Set $x(t)$, $t \in [kT, (k+1)T]$ as the solution of

$$\dot{x}(t) = f(\delta(q(kT), \sigma(kT)), x(t), u(kT), d(t - kT));$$

end for
return $(q(t), x(t))$, $t \in [0, NT]$.

By the above definition, the discrete state trajectory $q(\cdot)$ is piecewise constant with jumps occuring at sampling instants, while $x(\cdot)$ is a continuous function of time. Furthermore, it should be emphasized that although the control values are to be held constant on sampling intervals (as consistent with a sampled-data setting), the disturbance is allowed to choose a time-varying realization on each sampling interval, possibly in response to the control input selection. This allows

us to treat a range of robust control problems and differential game problems in which the noise or disturbance entering into the continuous dynamics may be adversarial under worst-case assumptions.

3.2.1 Example - Pairwise Aircraft Conflict Resolution

In the following, we will illustrate this modeling framework through an example of aircraft conflict resolution, as adapted from Tomlin et al. (2000); Mitchell et al. (2005); Hwang et al. (2005), in which it is used as a benchmark for hybrid and nonlinear reachability analysis. A similar model as presented here has been employed in Teo and Tomlin (2003) for an experimentally demonstrated conflict detection and resolution algorithm for closely-spaced parallel runway approaches.

The conflict scenario involves two aircraft moving in the plane, one of which is controlled (referred to as aircraft 1), while the other is uncontrolled (referred to as aircraft 2). The task is to synthesize the controls for aircraft 1 so as to avoid a collision with aircraft 2, subject to the worst-case controls of aircraft 2. The relative motion of aircraft 2 with respect to aircraft 1 is modeled using the following kinematic equations.

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} \begin{bmatrix} -u_1 + d_1 \cos x_3 + u_2 x_2 \\ d_1 \sin x_3 - u_2 x_1 \\ d_2 - u_2, \end{bmatrix} = \tilde{f}(x, u_1, u_2, d_1, d_2)$$

where x_1, x_2 is the relative position of aircraft 2 in the aircraft 1 reference frame; x_3 is the relative heading of aircraft 2 in the aircraft 1 reference frame; u_1, d_1 are the linear velocities of aircraft 1 and 2, respectivley; u_2, d_2 are the angular velocities of aircraft 1 and 2, respectively. Now consider a simplified model for the control system of aircraft 1 as represented by the state transition diagram shown in Figure 3.1.

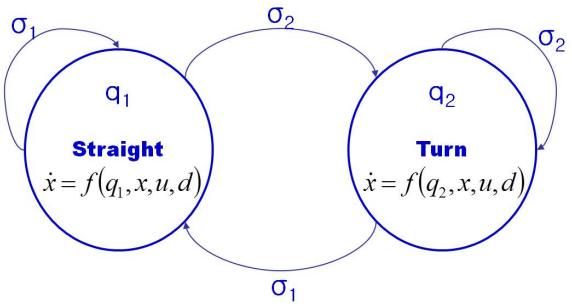


Figure 3.1: Two mode control system for aircraft conflict resolution.

For this particular example, the discrete state space is $Q = \{q_1, q_2\}$, where q_1 is a straight and level flight maneuver in which aircraft 1 modifies its linear velocity, while q_2 is a turning maneuver in which aircraft 1 modifies its angular velocity. The continuous state space within each discrete

state is $X = \mathbb{R}^3$. The set of discrete inputs is given by $\Sigma = \{\sigma_1, \sigma_2\}$, where σ_i corresponds to a switch to mode q_i. This gives rise to the discrete transition function

$$\delta(q,\sigma) = \begin{cases} q_1, & \sigma = \sigma_1 \\ q_2, & \sigma = \sigma_2. \end{cases}$$

In mode q_1, the linear velocity input u_1 takes on a range $U_1 = [\underline{v}, \overline{v}] \subset \mathbb{R}$, while the angular velocity is held constant at some value $u_2 = \omega_0$. In mode q_2, the angular velocity input u_2 takes on a range $U_2 = [\underline{\omega}, \overline{\omega}] \subset \mathbb{R}$, while the linear velocity is held constant at some value $u_1 = v_0$. This results in the continuous input space $U = U_1 \times U_2 \subset \mathbb{R}^2$. Similarly, the inputs d_1, d_2 of aircraft 2 are assumed to be chosen from compact sets $D_1, D_2 \subset \mathbb{R}$, resulting in the disturbance input space $D = D_1 \times D_2 \subset \mathbb{R}^2$. The corresponding vector fields in modes q_1 and q_2 are given by

$$f(q,x,u,d) = \begin{cases} \tilde{f}(x, u_1, \omega_0, d_1, d_2), & q = q_1 \\ \tilde{f}(x, v_0, u_2, d_1, d_2), & q = q_2. \end{cases}$$

3.3 Problem Formulations

Within the framework of sampled-data switched systems, we will consider two controller synthesis problems which commonly arise in safety-critical control applications. In the *safety* control problem, the objective is to synthesize a control policy μ, such that the closed-loop system trajectory $(q(\cdot), x(\cdot))$ remains within a prescribed safe set at all times. This could be, for example, a flight envelope protection problem for an aircraft. On the other hand, in the *reach-avoid* control problem, the objective is to synthesize a control policy, such that the closed-loop trajectory enters a target set within finite time while remaining outside an unsafe set. This could be, for example, an autonomous navigation problem, in which the target set is a goal region, while the unsafe set is comprised of the obstacles in the environment.

To be more precise, suppose we are given a sampled-data switched system $\mathscr{H}_{sw}$ and a collection of sets $W_i \subseteq R^n$ which specifies the safe set within each mode $q_i \in Q$. Then the set of safe states for $\mathscr{H}_{sw}$ is given by $W^H = \bigcup_{i=1}^m \{q_i\} \times W_i$. A formal statement of the safety control problem is as follows.

Problem 3.1. Given a sampled-data switched system $\mathscr{H}_{sw}$, time horizon $N > 0$, and safe set W^H:

1. Compute a set of states $G^N_{Safe} \subset Q \times X$ such that there exists an admissible control policy $\mu \in \mathscr{M}$ so that for any initial condition $(q_0, x_0) \in G^N_{Safe}$ and disturbance strategy $\gamma \in \Gamma$, the closed-loop state trajectory $(q(\cdot), x(\cdot))$, as defined by Algorithm 3.2.1, satisfies $(q(t), x(t)) \in W^H$ for all $t \in [0, NT]$;

2. Synthesize a control policy $\mu \in \mathscr{M}$ such that the above conditions are satisfied for each initial condition in G^N_{Safe}.

For the rest of this paper, we will refer to G^N_{Safe} as a *horizon-N safe set*, and a control policy which satisfies the safety specification on G^N_{Safe} as a *horizon-N safe control policy* with respect to G^N_{Safe}. By letting $N \to \infty$, we can consider an infinite horizon version of this problem in which the objective is to keep the system state within a set W^H for every $t \geq 0$. This may be of interest, for example, in a robust stabilization application with specifications of maintaining the closed-loop trajectory within a region around the origin.

Now suppose instead that we are given a target set $R^H = \bigcup_{i=1}^m \{q_i\} \times R_i$ that the system state is required to reach within some finite time horizon $[0, NT]$, and an avoid set $A^H = \bigcup_{i=1}^m \{q_i\} \times A_i$ that the system state is required to stay away from at all times. Then the reach-avoid problem can be formulated as follows.

Problem 3.2. Given a sampled-data switched system $\mathscr{H}_{sw}$, time horizon N, target set R^H and avoid set A^H:

1. Compute a set of states $G^N_{RA} \subset Q \times X$ such that there exists an admissible control policy $\mu \in \mathscr{M}$ so that for any initial condition $(q_0, x_0) \in G^N_{RA}$ and disturbance strategy $\gamma \in \Gamma$, the closed-loop state trajectory $(q(\cdot), x(\cdot))$, as defined by Algorithm 3.2.1, satisfies $(q(kT), x(kT)) \in R^H$ for some $k \in \{0, 1, \ldots, N\}$, and $(q(t), x(t)) \notin A^H$ for all $t \in [0, kT]$;

2. Synthesize a control policy $\mu \in \mathscr{M}$ such that the above conditions are satisfied for each initial condition in G^N_{RA}.

As before, we will refer to G^N_{RA} as a *horizon-N reach-avoid set*, and a control policy which satisfies the reach-avoid specification on G^N_{RA} as a *horizon-N reach-avoid control policy* with respect to G^N_{RA}.

Our approach to the safety and reach-avoid problems consists of first computing an approximate representation of the set G^N_{Safe} or G^N_{RA} through an iterative reachability algorithm, and then deriving a feedback control policy μ in terms of the collections of reachable sets returned by the reachability computation. In order to facilitate the computation of continuous reachable sets, as well as to ensure a finite representation of the feedback policy, we will impose the following set of assumptions.

Assumption 3.1.

1. The continuous input space U is discretized into a finite set $\tilde{U} \subset U$, called a quantized input set.

2. For each mode q_i, the unsafe set $W^C_i \subset X$ is closed and can be represented by a bounded and Lipschitz continuous function $\phi_{W^C_i} : X \to \mathbb{R}$ such that

$$W^C_i = \left\{ x \in X, \phi_{W^C_i}(x) \leq 0 \right\}.$$

3. For each mode q_i, the target set R_i and the avoid set A_i are closed and can be represented by bounded and Lipschitz continuous functions $\phi_{R_i} : X \to \mathbb{R}$ and $\phi_{A_i} : X \to \mathbb{R}$ such that

$$R_i = \{x \in X, \phi_{R_i}(x) \leq 0\},$$
$$A_i = \{x \in X, \phi_{A_i}(x) \leq 0\}.$$

In Assumption 3.1, a quantized input set is specified in order to obtain a finite representation of the control policy, in the form of a finite collection of reachable sets. In certain application scenarios, this may also have a practical significance in that, due to digital implementation or high level abstractions, the range of control choices may be quantized. Furthermore, when the continuous dynamics (3.1) is affine in the control input, the optimal controls for time-optimal steering problems, which have close relationships to reachability problems, can be sometimes chosen from within a finite set on the boundary of the input set (known as the bang-bang principle). In particular examples of nonlinear systems, it may also be possible to prove such properties using an optimal control argument Bayen et al. (2007). For problems of this type, the quantization levels can be chosen according to the set of optimal inputs.

On the other hand, the assumptions on the unsafe set, target set, and avoid set are necessary for the numerical computation of reachable sets through level set methods (Mitchell et al., 2005). The functions $\phi_{W_i^C}$, ϕ_{R_i}, and ϕ_{A_i} are commonly referred to as the level set representation of the sets W_i^C, R_i, and A_i, respectively. As an example, consider the pairwise aircraft conflict resolution scenario described in the previous section, and suppose that the collision zone is specified as a disc of radius r_0 centered on aircraft 1, then a level set representaion of the unsafe set W_i^C is simply given by

$$\phi_{W_i^C}(x) = \sqrt{x_1^2 + x_2^2} - r_0, \; i = 1, 2.$$

3.4 Safety Controller Synthesis

In this section, we discuss a solution approach to the safety control problem under Assumption 3.1. First, an algorithm is constructed for computing, in an offline setting, a set of feasible initial conditions for the safety problem using a restricted class of control policies whose range lies within the quantized input set. Second, it is shown that the result of this reachability computation gives a representation for a control policy satisfying the safety objective, and an algorithm is given for implementing this policy in an online setting. Towards the end of the section, we also discuss extensions of this methodology to the infinite horizon case.

For notational conveniences, we denote by $\tilde{\mathscr{M}} \subset \mathscr{M}$ the subset of control policies which selects continuous control inputs from the quantized input set $\tilde{U}$.

3.4.1 Safe Set Computation

Let $\mathscr{H}_{sw} = (Q, X, \Sigma, U, D, \delta, f, T)$ be a sampled-data switched system defined as in section 3.2. For a fixed $q_i \in Q$ and $\tilde{u} \in \tilde{U} \subset U$, consider a continuous state reachability operator $\mathscr{A}_T^{q_i,\tilde{u}}$, which takes

as its argument a set $G \subseteq X$ and produces as its output the set of states that can be forced inside G within $[0,T]$ by some realization of the disturbance.

$$\mathscr{A}_T^{q_i,\tilde{u}}(G) = \{x_0 \in X : \exists d(\cdot) \in \mathscr{D}_T, \exists t \in [0,T],\ x(t) \in G\}, \tag{3.2}$$

where $x(\cdot)$ is the solution of the ODE

$$\dot{x}(t) = f(q_i, x(t), \tilde{u}, d(t)),\ x(0) = x_0$$

on the interval $[0,T]$.

The computation of this set can be viewed within either an optimal control or a differential game framework. Under an optimal control interpretation, the disturbance is assumed to select a worst-case choice of realization $d(\cdot) \in \mathscr{D}_T$, in response to a fixed choice of control $u(t) = \tilde{u}$, $\forall t \in [0,T]$, so as to to drive the system state into G. On the other hand, this can be also viewed as a special case of a differential game on $[0,T]$ in which the control choice is restricted to the singleton $\tilde{u}$. Using either interpretation, one can characterize the evolution of the unsafe set through an appropriate Hamilton-Jacobi equation (Evans and Souganidis, 1984; Bardi and Capuzzo-Dolcetta, 1997; Mitchell et al., 2005).

In particular, suppose that the set G has a level set representation $\phi_G : X \to \mathbb{R}$. Let $\phi : X \times [-T,0] \to \mathbb{R}$ be the unique viscosity solution (Crandall and Lions, 1983) to the following HJB equation

$$\frac{\partial \phi}{\partial t} + \min\left[0, H\left(x, \frac{\partial \phi}{\partial x}\right)\right] = 0,\ \phi(x,0) = \phi_G(x), \tag{3.3}$$

where the optimal Hamiltonian is given by

$$H(x,p) = \min_{d \in D} p^T f(q_i, x, \tilde{u}, d). \tag{3.4}$$

Then by a special case of the arguments presented in Evans and Souganidis (1984) and Mitchell et al. (2005), we have

$$\mathscr{A}_T^{q_i,\tilde{u}}(G) = \{x \in X, \phi(x,-T) \leq 0\}.$$

Several remarks are in order. First, the minimization with respect to d in equation (3.4) gives the disturbance a slight advantage, as the disturbance has knowledge of the control input $\tilde{u}$ on $[0,T]$. Second, the Hamiltonian in equation (3.4) can be calculated analytically for systems in which the disturbance enters affinely in the model, namely when the vector field in each mode q_i can be written in the form $f(q_i,x,u,d) = \tilde{f}_i(x,u) + g_i(x)d$, and the disturbance input space takes the form $D = \prod_{i=1}^{n_d} [\underline{d}_i, \overline{d}_i]$. Note, however, that $\tilde{f}_i$ is not required to be affine in u. Third, the $\min[0,H]$ formulation in equation (3.3) constrains the reachable set to grow over time, which results in the property that $G \subseteq \mathscr{A}_T^{q_i,\tilde{u}}(G)$.

On the computational side, a numerical toolbox (Mitchell, 2007*a*) is available to compute a convergent approximation of the viscosity solution to (3.3) on a discrete grid of the continuous state space $X = \mathbb{R}^n$, based upon an implementation of level set methods (Sethian, 1999; Osher and Fedkiw, 2002). However, due to the fact that this grid is chosen to be uniform for numerical

convergence, the computational cost scales exponentially with the continuous state dimension, which currently limits the application of this method to problems with $n \leq 5$.

Now consider a discrete reachability operator *Reach*, taking as its argument a discrete state $q \in Q$ and producing as its output the subset of Q reachable from q in one step:

$$Reach(q) := \left\{ q' \in Q : \exists \sigma \in \Sigma,\ \delta(q,\sigma) = q' \right\}.$$

It can be inferred in a straightforward manner that this operator can be computed as $Reach(q) = \bigcup_{\sigma \in \Sigma} \delta(q,\sigma)$. By the definition of Σ, this union is finite.

Given a set $G^H = \bigcup_{i=1}^m \{q_i\} \times G_i \subset Q \times X$, we define the *one-step unsafe set* with respect to G^H as follows.

$$\begin{aligned} \mathscr{A}_T^H(G^H) = \big\{ (q,x) \in Q \times X : \forall (\sigma,\tilde{u}) \in \Sigma \times \tilde{U}, \exists d \in \mathscr{D}_T, \\ \exists t \in [0,T],\ (q(t),x(t)) \in G^H \big\}, \end{aligned} \tag{3.5}$$

In other words, this is the set of initial conditions which can reach G^H within one time step under an admissible disturbance realization, regardless of the choice of control inputs. The following result provides a representation for $\mathscr{A}_T^H$ in terms of the continuous reachability operator $\mathscr{A}_T^{q_i,\tilde{u}}$ and the discrete reachability operator *Reach*.

Lemma 3.1. *Let $G^H = \bigcup_{i=1}^m \{q_i\} \times G_i \subset Q \times X$. Then*

$$\mathscr{A}_T^H(G^H) = \bigcup_{q_i \in Q} \{q_i\} \times G_i \cup \left(\bigcap_{q_j \in Reach(q_i)} \bigcap_{\tilde{u} \in \tilde{U}} \mathscr{A}_T^{q_j,\tilde{u}}(G_j) \right). \tag{3.6}$$

Proof. For notational conveniences, we define $V^H = \bigcup_{i=1}^m \{q_i\} \times G_i \cup V_i$, where

$$V_i = \bigcap_{q_j \in Reach(q_i)} \bigcap_{\tilde{u} \in \tilde{U}} \mathscr{A}_T^{q_j,\tilde{u}}(G_j).$$

Let $(q_i,x) \in (\mathscr{A}_T^H(G^H))^C$. By the definition in (3.5), there exists a choice of controls $(\sigma,\tilde{u}) \in \Sigma \times \tilde{U}$ such that for every choice of disturbance realization $d \in \mathscr{D}_T$, the one step trajectory of $\mathscr{H}_{sw}$ initialized at (q_i,x) satisfies $(q(t),x(t)) \notin G^H$, $\forall t \in [0,T]$. Let $q_j = \delta(q_i,\sigma)$, then this implies that $x \notin G_i$ and $x \notin \mathscr{A}_T^{q_j,\tilde{u}}(G_j)$, and hence $x \notin G_i \cup V_i$. Thus, we have $(\mathscr{A}_T^H(G^H))^C \subseteq (V^H)^C$, or equivalently, $V^H \subseteq \mathscr{A}_T^H(G^H)$.

In order to prove the reverse inclusion, consider a state $(q_i,x) \in (V^H)^C$. Then by the definition of V^H, $x \notin G_i$ and there exists $q_j \in Reach(q_i)$ and $\tilde{u} \in \tilde{U}$ such that $x \notin \mathscr{A}_T^{q_j,\tilde{u}}(G_j)$. Let $\sigma \in \Sigma$ be a discrete command such that $q_j = \delta(q_i,\sigma)$. Then under the choice of controls $(\sigma,\tilde{u})$, the trajectory of $\mathscr{H}_{sw}$ starting from (q_i,x) satisfies $(q(t),x(t)) \notin G^H$, $\forall t \in [0,T]$, regardless of any admissible disturbance realization. Thus, $(q_i,x) \notin A_T^H(G^H)$, from which it follows that $(V^H)^C \subseteq (\mathscr{A}_T^H(G^H))^C$, or equivalently, $\mathscr{A}_T^H(G^H) \subseteq V^H$. □

We note briefly that under level set representations, computing the union or intersection of sets reduces to computing the pointwise minimum or maximum of level set functions. Specifically, suppose ϕ_A and ϕ_B are level set representations of sets A and B, respectively, then the set $A \cup B$ is represented by $\min\{\phi_A, \phi_B\}$.

Now consider Algorithm 3.4.1 for computing a horizon-N unsafe set under the policy class $\tilde{\mathscr{M}}$.

Algorithm 3.4.1 Computation of horizon-N Unsafe Set

Require: $W^H \subset Q \times X$ and $N \geq 1$
1: $V_0^H \Leftarrow (W^H)^C$
2: **for** $j = 0$ to $N-1$ **do**
3: $\quad V_{j+1}^H \Leftarrow \mathscr{A}_T^H(V_j^H)$
4: **end for**
5: **return** $V_1^H, V_2^H, ..., V_N^H$

Proposition 3.1. *Given a sampled-data switched system $\mathscr{H}_{sw}$ and a safe set $W^H \subset Q \times X$, let V_N^H be the output of Algorithm 3.4.1. Then $(V_N^H)^C$ is a horizon-N safe set.*

Proof. Given $\tilde{\mu} \in \tilde{\mathscr{M}}$ and $\gamma \in \Gamma$, we denote by $\tilde{\mu}_{k\to N}$ the sequence $(\tilde{\mu}_k, \tilde{\mu}_{k+1}, ..., \tilde{\mu}_{N-1})$, and by $\gamma_{k\to N}$ the sequence $(\gamma_k, \gamma_{k+1}, ..., \gamma_{N-1})$. The corresponding truncated control policy space and disturbance strategy space are denoted by $\tilde{\mathscr{M}}_{k\to N}$ and $\Gamma_{k\to N}$, respectively. We will prove the following statement by backward induction on k: there exists a control policy $\tilde{\mu}_{k\to N} \in \tilde{\mathscr{M}}_{k\to N}$ such that for every initial condition $(q_k, x_k) \in (V_{N-k}^H)^C$ and disturbance strategy $\gamma_{k\to N} \in \Gamma_{k\to N}$, the closed-loop trajectory of $\mathscr{H}_{sw}$ satisfies $(q(t), x(t)) \in W^H$, $\forall t \in [kT, NT]$. Clearly, the statement of the proposition follows from the case of $k = 0$.

First, for the case of $k = N-1$, we have $V_1^H = \mathscr{A}_T^H((W^H)^C)$. By the definition of $\mathscr{A}_T^H$ in (3.5), for every $(q,x) \in (V_1^H)^C$, there exists a choice of control input $(\sigma, \tilde{u})_{(q,x)} \in \Sigma \times \tilde{U}$ such that for every disturbance realization $d \in \mathscr{D}_T$, the one step state trajectory satisfies $(q(t), x(t)) \in W^H$, $\forall t \in [(N-1)T, NT]$. Let $\tilde{\mu}_{N-1}^*(q,x) = (\sigma, \tilde{u})_{(q,x)}$, $\forall (q,x) \in (V_1^H)^C$, then $\tilde{\mu}_{N-1}^*$ is a safe control policy with respect to $(V_1^H)^C$. Second, for the inductive step, we assume that for some $j \in \{1, 2, ..., N-1\}$, there exists a safe control policy $\tilde{\mu}_{j\to N} \in \tilde{\mathscr{M}}_{j\to N}$ with respect to $(V_{N-j}^H)^C$. On the set $(V_{N-j+1}^H)^C = (\mathscr{A}_T^H(V_{N-j}^H))^C$, choose a one step control policy $\tilde{\mu}_{j-1}^*$ such that the trajectory of $\mathscr{H}_{sw}$ over $[(j-1)T, jT]$ avoids the set V_{N-j}^H (the existence of such a policy is again implied by (3.5)). Then $\tilde{\mu}_{j-1\to N} = (\tilde{\mu}_{j-1}^*, \tilde{\mu}_{j\to N})$ is a safe control policy with respect to $(V_{N-j+1}^H)^C$. The result then follows by induction. □

3.4.2 Safe Control Policies

In the proof of Proposition 3.1, we showed the existence a safe control policy with respect to $(V_N^H)^C$ within the restricted policy class $\tilde{\mathscr{M}}$. The question then becomes whether an explicit representation

for such a policy can be derived. It turns out that the reachable sets generated by Algorithm 3.4.1 provide us with such a representation.

Motivated by the expression for the reachability operator $\mathscr{A}_T^H$ in (3.6), we construct the following set-valued feedback maps for the choice of safe control inputs:

$$F_k^{Safe}(q,x) = \left\{(\sigma,\tilde{u}) \in \Sigma \times \tilde{U} : x \notin \mathscr{A}_T^{\delta(q,\sigma),\tilde{u}}(V_{N-k-1}^H(\delta(q,\sigma)))\right\}, \tag{3.7}$$

for $(q,x) \in (V_{N-k}^H)^C$ and $k = 0,1,...,N-1$. In the above, we denote by $V_j^H(q_i)$ the component of V_j^H in mode q_i. The following result provides us with a formal proof that these set-valued maps indeed constitute a finite horizon safe control policy on $(V_N^H)^C$.

Proposition 3.2. *Let V_j^H, $j = 1,...,N$ be the j-step unsafe sets, as computed using Algorithm 3.4.1. If $(V_N^H)^C \neq \emptyset$, then any control policy $\tilde{\mu} \in \tilde{\mathscr{M}}$ which satisfies*

$$\tilde{\mu}_k(q,x) = F_k^{Safe}(q,x),\ \forall (q,x) \in (V_{N-k}^H)^C, \tag{3.8}$$

for $k = 0,1,...,N-1$ is a horizon-N safe control policy with respect to $(V_N^H)^C$.

Proof. By the representation of the reachability operator $\mathscr{A}_T^H$ in (3.6), it can be inferred that the sets V_j^H satisfy

$$V_0^H \subseteq V_1^H \subseteq \cdots \subseteq V_N^H.$$

Thus, $(V_N^H)^C \neq \emptyset$ implies $(V_j^H)^C \neq \emptyset$, $\forall j = 0,1,...,N$. Furthermore, given that $V_0^H = (W^H)^C$, we also have $(V_j^H)^C \subseteq W^H$, $\forall j = 0,1,...,N$.

Let $\tilde{\mu} \in \tilde{\mathscr{M}}$ be any control policy which satisfies (3.8). We prove the following statement by forward induction on k: for any initial condition $(q(0),x(0)) \in (V_N^H)^C$ and disturbance strategy $\gamma \in \Gamma$, the trajectory of $\mathscr{H}_{sw}$ satisfies $(q(t),x(t)) \in W^H$, $\forall t \in [0,kT]$ and $(q(kT),x(kT)) \in (V_{N-k}^H)^C$. The proposition follows from the case of $k = N$.

For $k = 0$, it is clear that $(q(0),x(0)) \in (V_N^H)^C \subseteq W^H$. For the inductive step, we assume that for some $j \in \{0,1,...,N-1\}$, the system trajectory satisfies $(q(t),x(t)) \in W^H$, $\forall t \in [0,jT]$ and $(q(jT),x(jT)) \in (V_{N-j}^H)^C$, regardless of the disturbance strategy $\gamma \in \Gamma$. With the assumption on $\tilde{\mu}$, we have $\tilde{\mu}_j(q(jT),x(jT)) = F_j^S(q(jT),x(jT))$. From (3.2) and (3.7), it can be then inferred that for any control input $(\sigma,\tilde{u}) \in \tilde{\mu}_j(q(jT),x(jT))$, the one-step trajectory satisfies $(q(t),x(t)) \in (V_{N-j-1}^H)^C \subseteq W^H$, $\forall t \in [jT,(j+1)T]$, regardless of the disturbance realization. The result then follows by induction. □

Using this result, we can compute using Algorithm 3.4.1 the collections of level set functions representing the sets V_j^H in an offline setting, and then use these functions as lookup tables to extract safe control inputs as state measurements are received. A possible implementation of this procedure is given in Algorithm 3.4.2.

It should be remarked that given level set representations $\phi_k^{q',\tilde{u}}$ of the sets $\mathscr{A}_T^{q',\tilde{u}}(V_{N-k-1}^H(q'))$, checking the condition $x(kT) \notin \mathscr{A}_T^{q',\tilde{u}}(V_{N-k-1}^H(q'))$ is equivalent to checking the condition

$$\phi_k^{q',\tilde{u}}(x(kT)) > 0.$$

Algorithm 3.4.2 Online Implementation of Finite Horizon Safe Control Policy

Require: V_j^H, $j = 1,...,N$, $(q(0),x(0)) \in (V_N^H)^C$

for $k = 0$ to $N-1$ **do**
  $F_k^{Safe} \Leftarrow \emptyset$;
  Measure state $(q(kT),x(kT))$;
  for all $(\sigma,\tilde{u}) \in \Sigma \times \tilde{U}$ **do**
    $q' \Leftarrow \delta(q(kT),\sigma)$;
    if $x(kT) \notin \mathscr{A}_T^{q',\tilde{u}}(V_{N-k-1}^H(q'))$ **then**
      Add $(\sigma,\tilde{u})$ to F_k^{Safe};
    end if
  end for
  Apply input $(\sigma_k,\tilde{u}_k) \in F_k^{Safe}$
end for

3.4.3 Infinite Horizon Safety Problem

Now consider an extension of the finite horizon safety control problem as discussed in the preceding sections to the case in which the control objective is to keep the system trajectory within a safe set W^H for all times. Specifically, we are interested in computing a set $G_{Safe}^{\infty} \subset Q \times X$ such that there exists an admissible control policy $\mu \in \mathscr{M}$ so that for any initial condition $(q_0,x_0) \in G_{Safe}^{\infty}$ and disturbance strategy $\gamma \in \Gamma$, the closed-loop state trajectory $(q(\cdot),x(\cdot))$ satisfies $(q(t),x(t)) \in W^H$ for all $t \geq 0$. Furthermore, we would like to derive an infinite horizon safe control policy from the result of such a computation.

As observed in the proof of Proposition 3.2, the sequence of unsafe sets V_j^H, $j = 0,1,\cdots$, as computed by Algorithm 3.4.1 satisfies the following monotonicity condition:

$$V_0^H \subseteq V_1^H \subseteq V_2^H \subseteq \cdots.$$

It is then intuitive that if V_j^H gradually stops growing with successive iterations of the algorithm and converges to a maximal unsafe set V_{∞}^H, the set of all states which lie outside V_{∞}^H is an infinite horizon safe set.

More precisely, suppose that Algorithm 3.4.1 converges to a fixed point of the operator $\mathscr{A}_T^H$ within a finite number of iterations, namely

$$V_{N_0+1}^H = \mathscr{A}_T^H(V_{N_0}^H) = V_{N_0}^H, \text{ for some } N_0 < \infty. \tag{3.9}$$

Then by induction, it can be inferred that $V_N^H = V_{N_0}^H$, $\forall N \geq N_0$. Applying Proposition 3.1, it then follows that $(V_{N_0}^H)^C$ is a horizon-N safe set for every $N \geq N_0$, and hence an infinite horizon safe set. In the case that this set is nonempty, we can also derive an infinite horizon safe control policy from the representation of $V_{N_0}^H$. Specifically, consider the set of safe control inputs defined by

$$F^{Safe}(q,x) = \left\{(\sigma,\tilde{u}) \in \Sigma \times \tilde{U} : x \notin \mathscr{A}_T^{\delta(q,\sigma),\tilde{u}}(V_{N_0}^H(\delta(q,\sigma)))\right\}. \tag{3.10}$$

Then by a similar argument as in the proof of Proposition 3.2, any stationary policy $\tilde{\mu} = (\tilde{\mu}, \tilde{\mu}, ...) \in \tilde{\mathscr{M}}$ which satisfies

$$\tilde{\mu}(q,x) = F^{Safe}(q,x),\ \forall (q,x) \in (V_{N_0}^H)^C$$

is an infinite horizon safe control policy with respect to $(V_{N_0}^H)^C$. An algorithm for implementing such a control policy can be constructed similarly as Algorithm 3.4.2.

In the literature, conditions under which (3.9) holds have been studied in terms of the concept of *decidability*, which in this case concerns whether an infinite horizon reachability question can be answered by a finite computation. To the best of our knowledge, currently known decidability results for hybrid systems reachability are restricted to the class of timed automata, linear hybrid automata, and linear continuous dynamics with special structures (Henzinger et al., 1998; Alur et al., 2000). Nonetheless, for certain classes of problems in nonlinear differential games, it has been shown that a maximal unsafe set exists (Isaacs, 1967; Merz, 1972), and that a numerical reachability computation indeed converges to such a set (Mitchell et al., 2005). In such cases, one may check (3.9) in terms of the convergence of the level set functions representing V_j^H.

Revisiting the aircraft conflict resolution example from Section 3.2, consider a safety control problem where we would like to keep the relative aircraft states away from a collision zone as defined by

$$W_i^C = \left\{ x \in \mathbb{R}^3 : \sqrt{x_1^2 + x_2^2} \leq r_0 \right\},\ i = 1,2$$

for some positive radius $r_0 > 0$. For the reachability computation, we select the input bounds for aircraft 1 and aircraft 2 as follows: in mode 1, the velocity range of aircraft 1 is chosen to be $U_1 = [400\text{ kts}, 500\text{ kts}]$, with a constant heading input $\omega_0 = 0$; in mode 2, the velocity of aircraft 1 is fixed at $v_0 = 450$ kts, while the angular velocity is allowed to vary within the range $U_2 = [-2\text{ deg/s}, 2\text{ deg/s}]$; in both modes, the aircraft 2 input ranges are chosen to be $D_1 = [400\text{ kts}, 500\text{ kts}]$ and $D_2 = [-1\text{ deg/s}, 1\text{ deg/s}]$. The collision zone radius is set as $r_0 = 5$ nmi, while the sampling interval is set as $T = 10$ sec.

With a uniform discretization of U_1 and U_2 into 11 input levels, we perform an unsafe set computation using Algorithm 3.4.1. In this case, it was found that this computation converges to within numerical accuracy of a fixed point after about 7 time steps. The resulting infinite horizon unsafe set $(G_{Safe}^{\infty})^C$ is shown in Figure 3.2 along with the collision zone $(W^H)^C$. To illustrate the set-valued control policy obtained from this computation, we take a slice of the unsafe sets $\mathscr{A}_T^{q_1,500}((G_{Safe}^{\infty}(q_1))^C)$ and $\mathscr{A}_T^{q_2,2}((G_{Safe}^{\infty}(q_2))^C)$ at a relative heading angle of π radians (a scenario in which the two aircraft are directly facing each other). According to (3.10), in the complement of the set $\mathscr{A}_T^{q_1,500}((G_{Safe}^{\infty}(q_1))^C)$, one can choose the straight maneuver $(\sigma_1, 500\text{ kts})$ as the safe input, while in the complement of $\mathscr{A}_T^{q_2,2}((G_{Safe}^{\infty}(q_2))^C)$, one can choose the turn maneuver $(\sigma_2, 2\text{ deg/s})$.

From the result of this reachability computation, the infinite horizon safety controller is synthesized and implemented in simulation using Algorithm 3.4.2, with aircraft 2 applying random inputs chosen from within its input ranges D_1 and D_2. A sample run of this simulation is given in Figure 3.3, in which aircraft 1 successfully avoids a collision with aircraft 2 over a 4 minute time

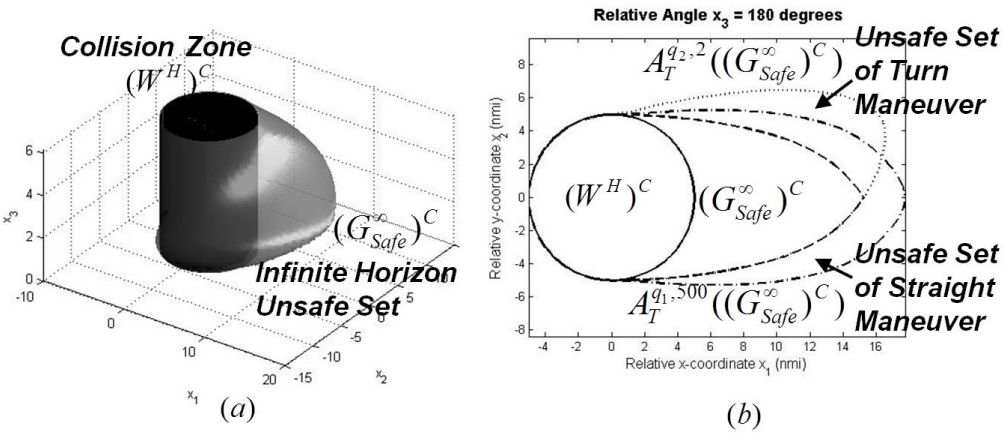


Figure 3.2: Results of infinite horizon reachability calculations for two aircraft conflict resolution example: (a) Infinite horizon unsafe set; (b) Slice of unsafe set at relative angle of π radians.

horizon. In this case, we ran a MATLAB implementation of Algorithm 3.4.2 on a 2 GHz Intel Xeon processor with 4 GB of memory, and the average computation time for each iteration of the algorithm was found to be approximately 0.1 seconds.

3.5 Reach-avoid Controller Synthesis

Using a similar approach as in the safety problem, we now discuss a solution to the finite horizon reach-avoid problem under Assumption 3.1. In particular, a reachability algorithm is given for computing the set of states reachable to a target set R^H while avoiding an unsafe set A^H, under the quantized policy space $\tilde{\mathscr{M}}$, along with a procedure for synthesizing a reach-avoid control policy from the result of this computation.

3.5.1 Reach-avoid Set Computation

For the objective of reaching a target set, we introduce a continuous state reachability operator $\mathscr{R}^{q_i,\tilde{u}}$, taking as its argument a set $G \subseteq X$ and producing as its output the set of states that can be controlled inside G at time T, regardless of the disturbance realization:

$$\mathscr{R}_T^{q_i,\tilde{u}}(G) = \{x_0 \in X : \forall d(\cdot) \in \mathscr{D}_T,\ x(T) \in G\},$$

where $x(\cdot)$ is the solution of the ODE $\dot{x}(t) = f(q_i, x(t), \tilde{u}, d(t)),\ x(0) = x_0$ on the interval $[0,T]$.

The computation of $\mathscr{R}_T^{q_i,\tilde{u}}(G)$ can be also viewed from a differential game perspective, in which the control chooses an input $u(t) = \tilde{u},\ \forall t \in [0,T]$ so as to achieve $x(T) \in G$, while the disturbance selects, in response, a realization $d(\cdot) \in \mathscr{D}_T$ so as to prevent the control from doing so. An HJB

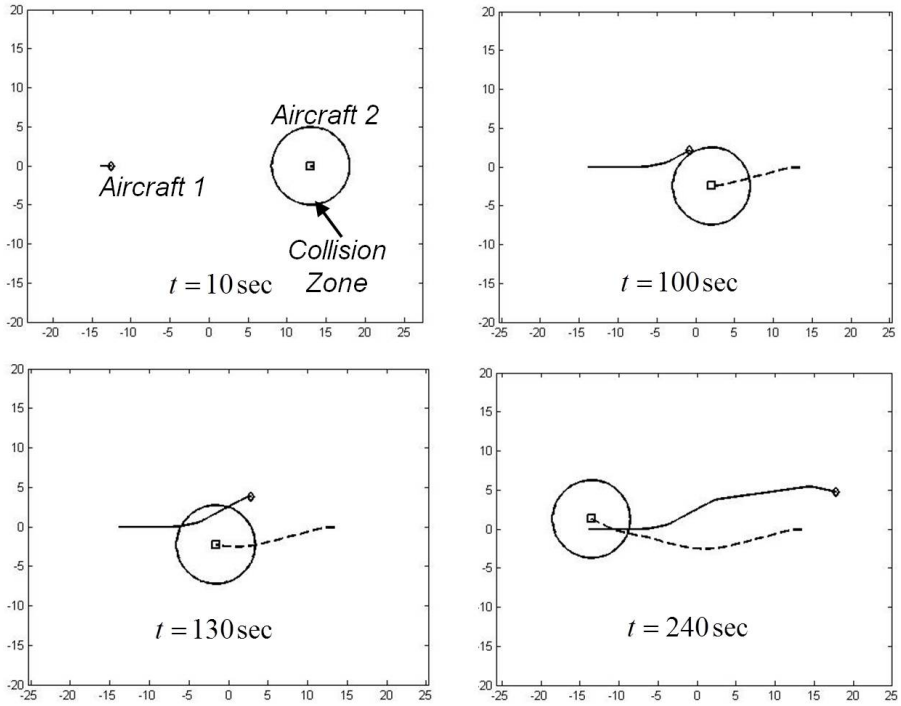


Figure 3.3: Sample simulation run of two aircraft conflict resolution example.

equation encoding this terminal cost problem is given by

$$\frac{\partial \phi}{\partial t} + H\left(x, \frac{\partial \phi}{\partial x}\right) = 0,\ \phi(x,0) = \phi_G(x), \tag{3.11}$$

with the optimal Hamiltonian

$$H(x,p) = \max_{d \in D} p^T f(q_i, x, \tilde{u}, d). \tag{3.12}$$

Let ϕ be the unique viscosity solution to (3.11), then by an application of the results in Evans and Souganidis (1984), it follows that

$$\mathscr{R}_T^{q_i,\tilde{u}}(G) = \{x \in X, \phi(x,-T) \leq 0\}.$$

Given a target set $G_1 \subset X$ and an avoid set $G_2 \subset X$, consider a one step reach-avoid operator for mode q_i under input $\tilde{u}$ as defined by

$$\mathscr{RA}_T^{q_i,\tilde{u}}(G_1,G_2) = \{x_0 \in X : \forall d(\cdot) \in \mathscr{D}_T, (x(T) \in G_1) \wedge (x(t) \notin G_2,\ \forall t \in [0,T])\}.$$

From the definitions of the operators $\mathscr{A}_T^{q_i,\tilde{u}}$ and $\mathscr{R}_T^{q_i,\tilde{u}}$, it can be inferred in a straightforward manner that

$$\mathscr{R}\mathscr{A}_T^{q_i,\tilde{u}}(G_1,G_2) = \mathscr{R}_T^{q_i,\tilde{u}}(G_1) \cap (\mathscr{A}_T^{q_i,\tilde{u}}(G_2))^C.$$

Now consider a one step reach-avoid operator for the switched system $\mathscr{H}_{sw}$ as defined by

$$\begin{aligned}\mathscr{R}\mathscr{A}_T^H(G_1^H,G_2^H) = \big\{(q,x) : \exists(\sigma,\tilde{u}) \in \Sigma\times\tilde{U}, \forall d \in \mathscr{D}_T, \\ ((q(T),x(T)) \in G_1^H) \wedge ((q(t),x(t)) \notin G_2^H,\ \forall t \in [0,T])\big\},\end{aligned} \tag{3.13}$$

where $G_1^H = \bigcup_{i=1}^m \{q_i\}\times G_{1,i}$ and $G_2^H = \bigcup_{i=1}^m \{q_i\}\times G_{2,i}$ are subsets of $Q\times X$. In other words, this is the set of states reachable to G_1^H at the end of a sampling interval, while avoiding G_2^H throughout. The following result provides a charaterization of this operator in terms of a combination of discrete and continuous reachability computations.

Lemma 3.2. *Let $G_1^H = \bigcup_{i=1}^m \{q_i\}\times G_{1,i} \subset Q\times X$ and $G_2^H = \bigcup_{i=1}^m \{q_i\}\times G_{2,i} \subset Q\times X$. Then*

$$\mathscr{R}\mathscr{A}_T^H(G_1^H,G_2^H) = \bigcup_{q_i\in Q} \{q_i\}\times(G_{2,i})^C \cap \left(\bigcup_{q_j\in Reach(q_i)} \bigcup_{\tilde{u}\in\tilde{U}} \mathscr{R}\mathscr{A}_T^{q_j,\tilde{u}}(G_{1,j},G_{2,j})\right). \tag{3.14}$$

Proof. For notational conveniences, we define $V^H = \bigcup_{i=1}^m \{q_i\}\times(G_{2,i})^C \cap V_i$, where

$$V_i = \bigcup_{q_j\in Reach(q_i)} \bigcup_{\tilde{u}\in\tilde{U}} \mathscr{R}\mathscr{A}_T^{q_j,\tilde{u}}(G_{1,j},G_{2,j}).$$

Let $(q_i,x) \in \mathscr{R}\mathscr{A}_T^H(G_1^H,G_2^H)$. By the definition in (3.13), there exists a choice of controls $(\sigma,\tilde{u}) \in \Sigma\times\tilde{U}$ such that for every choice of disturbance realization $d\in\mathscr{D}_T$, the one step trajectory of $\mathscr{H}_{sw}$ initialized at (q_i,x) satisfies $(q(T),x(T)) \in G_1^H$ and $(q(t),x(t)) \notin G_2^H$, $\forall t\in[0,T]$. Let $q_j = \delta(q_i,\sigma)$, then this implies that $x \notin G_{2,i}$ and $x \in \mathscr{R}\mathscr{A}_T^{q_j,\tilde{u}}(G_{1,j},G_{2,j})$, and hence $x \in (G_{2,i})^C \cap V_i$. Thus, we have $\mathscr{R}\mathscr{A}_T^H(G_1^H,G_2^H) \subseteq V^H$.

In order to prove the reverse inclusion, consider a state $(q_i,x) \in V^H$. Then by the definition of V^H, $x \notin G_{2,i}$ and there exists $q_j \in Reach(q_i)$ and $\tilde{u}\in\tilde{U}$ such that $x \in \mathscr{R}\mathscr{A}_T^{q_j,\tilde{u}}(G_{1,j},G_{2,j})$. Let $\sigma\in\Sigma$ be a discrete command such that $q_j = \delta(q_i,\sigma)$. Then under the choice of controls $(\sigma,\tilde{u})$, the trajectory of $\mathscr{H}_{sw}$ starting from (q_i,x) satisfies $(q(T),x(T)) \in G_1^H$ and $(q(t),x(t)) \notin G_2^H$, $\forall t\in[0,T]$, regardless of any admissible disturbance realization. Thus, $(q_i,x) \in \mathscr{R}\mathscr{A}_T^H(G_1^H,G_2^H)$, from which it follows that $V^H \subseteq \mathscr{R}\mathscr{A}_T^H(G_1^H,G_2^H)$. □

Now consider Algorithm 3.5.1 for computing a horizon-N reach-avoid set under quantized control policies.

Proposition 3.3. *Given a sampled-data switched system $\mathscr{H}_{sw}$, a target set $R^H \subset Q\times X$ and an avoid set $A^H \subset Q\times X$, let S_N^H be the output of Algorithm 3.5.1. Then S_N^H is a horizon-N reach-avoid set.*

Algorithm 3.5.1 Computation of Finite Horizon Reach-avoid Set

Require: $R^H, A^H \subset Q \times X$
1: $S_0^H \Leftarrow R^H \setminus A^H$
2: **for** $j = 0$ to $N-1$ **do**
3: $\quad S_{j+1}^H \Leftarrow \mathscr{R}\mathscr{A}_T^H(S_j^H, A^H) \cup S_j^H$;
4: **end for**
5: **return** S_1^H, S_2^H, ..., S_N^H

Proof. Similarly as in the proof of Proposition 3.1, we will prove the following statement by backwards induction on k: there exists $\tilde{\mu}_{k\to N} \in \tilde{\mathscr{M}}_{k\to N}$ such that for every initial condition $(q,x) \in S_{N-k}^H \setminus R^H$ and $\gamma_{k\to N} \in \Gamma_{k\to N}$, the trajectory of $\mathscr{H}_{sw}$ satisfies $(q((k+1)T), x((k+1)T)) \in S_{N-k}^H$, $(q(jT), x(jT)) \in R^H$ for some $j \in \{k+1, ..., N\}$ and $(q(t), x(t)) \notin A^H$, $\forall t \in [kT, jT]$. The statement of the proposition again follows from the case of $k = 0$.

First, for $k = N-1$, we have $S_1^H = \mathscr{R}\mathscr{A}_T^H(S_0^H, A^H) \cup S_0^H$, where $S_0^H = R^H \setminus A^H$. By the definition of $\mathscr{R}\mathscr{A}_T^H$ in (3.13), for every $(q,x) \in \mathscr{R}\mathscr{A}_T^H(S_0^H, A^H)$, there exists a choice of controls $(\sigma, \tilde{u})_{(q,x)} \in \Sigma \times \tilde{U}$ such that for every disturbance realization $d \in \mathscr{D}_T$, the closed-loop trajectory satisfies $(q(NT), x(NT)) \in R^H$ and $(q(t), x(t)) \notin A^H$, $\forall t \in [(N-1)T, NT]$. Let $\tilde{\mu}_{N-1}^*$ be any one step policy which satisfies $\tilde{\mu}_{N-1}^*(q,x) = (\sigma, \tilde{u})_{(q,x)}$, $\forall (q,x) \in \mathscr{R}\mathscr{A}_T^H(S_0^H, A^H)$, then μ_{N-1}^* has the required properties.

Next, suppose that the induction hypothesis holds for some $j \in \{1, 2, ..., N-1\}$. Then there exists a reach-avoid control policy $\tilde{\mu}_{j\to N} = (\tilde{\mu}_j, \tilde{\mu}_{j+1}, ..., \tilde{\mu}_{N-1})$ with respect to S_{N-j}^H. Furthermore, under the one step policy $\tilde{\mu}_j$, the closed-loop trajectory starting from any initial condition in $S_{N-j}^H \setminus R^H$ satisfies $(q((j+1)T), x((j+1)T)) \in S_{N-j}^H$. Now consider an initial condition $(q,x) \in \mathscr{R}\mathscr{A}_T^H(S_{N-j}^H, A^H)$. By (3.13), there exists a choice of controls $(\sigma, \tilde{u})_{(q,x)} \in \Sigma \times \tilde{U}$ such that the one step trajectory of $\mathscr{H}_{sw}$ satisfies $(q((j+1)T), x((j+1)T)) \in S_{N-j}^H$ and $(q(t), x(t)) \notin A^H$, $\forall t \in [jT, (j+1)T]$. Choose a one step policy $\tilde{\mu}_{j-1}^*$ as follows:

$$\tilde{\mu}_{j-1}^*(q,x) = \begin{cases} \tilde{\mu}_j(q,x), & (q,x) \in S_{N-j}^H \\ (\sigma, \tilde{u})_{(q,x)}, & (q,x) \in S_{N-j+1}^H \setminus S_{N-j}^H. \end{cases}$$

Then $\tilde{\mu}_{j-1\to N} = (\tilde{\mu}_{j-1}^*, \tilde{\mu}_{j\to N})$ is a control policy with the required properties. The desired result then follows by induction. □

3.5.2 Reach-avoid Control Policy

As in the case of the safety control problem, one can derive an explicit representation of the reach-avoid control policy from the reachability computation in Algorithm 3.5.1. In particular, suppose that $S_0^H = R^H \setminus A^H \neq \emptyset$, we define a function $k_{\min} : S_N^H \to \{0, 1, ..., N\}$ by

$$k_{\min}(q,x) = \min \left\{ j \in \{0, 1, ..., N\} : (q,x) \in S_j^H \right\}.$$

This can be interpreted as the minimum time to reach at a feasible initial condition $(q,x) \in S_N^H$, with respect to the quantized control policy space $\tilde{\mathscr{M}}$.

At a state $(q,x) \in S_N^H \setminus S_0^H$, consider the set of feasible control inputs for the reach-avoid problem as defined by

$$F^{RA}(q,x) = \Big\{(\sigma,\tilde{u}) \in \Sigma \times \tilde{U} : \\ x \in \mathscr{RA}_T^{\delta(q,\sigma),\tilde{u}}(S^H_{k_{\min}(q,x)-1}(\delta(q,\sigma)), A^H(\delta(q,\sigma)))\Big\} \tag{3.15}$$

It can be checked in a straightforward manner that this set is nonempty for every $(q,x) \in S_N^H \setminus S_0^H$. For $(q,x) \in S_0^H$, we define $F^{RA}(q,x) = \Sigma \times \tilde{U}$.

Proposition 3.4. *Let S_j^H, $j = 1,...,N$ be the j-step reach-avoid sets, as computed through Algorithm 3.5.1. If $R^H \setminus A^H \neq \emptyset$, then any control policy $\tilde{\mu} \in \tilde{\mathscr{M}}$ which satisfies*

$$\tilde{\mu}_k(q,x) = F^{RA}(q,x),\ \forall (q,x) \in S_N^H, \tag{3.16}$$

for $k = 0,1,...,N-1$, is a horizon-N reach-avoid control policy with respect to S_N^H.

Proof. Let $\tilde{\mu} \in \tilde{\mathscr{M}}$ be any control policy which satisfies (3.16). We prove the following statement by forward induction on k: for any initial condition $(q(0),x(0)) \in S_N^H$ and disturbance strategy $\gamma \in \Gamma$, the trajectory of $\mathscr{H}_{sw}$ on $[0,kT]$ satisfies at least one of the following conditions:

1. $\exists l \leq k$, $(q(lT),x(lT)) \in S_0^H$ and $(q(t),x(t)) \notin A^H$, $\forall t \in [0,lT]$;
2. $(q(kT),x(kT)) \in S_{N-k}^H$ and $(q(t),x(t)) \notin A^H$, $\forall t \in [0,kT]$.

The proposition then follows from the case of $k = N$.

Let $(q(0),x(0)) \in S_N^H$ and $\gamma \in \Gamma$. For $k = 0$, we have by the definition of the operator $\mathscr{RA}_T^H$ in (3.13) and the set S_0^H in Algorithm 3.5.1 that $S_N^H \subseteq (A^H)^C$. For the inductive step, we assume that either condition 1 or condition 2 holds for some $j \in \{0,1,...,N-1\}$. If condition 1 holds for the trajectory on $[0,jT]$, then clearly this condition also holds for the trajectory on $[0,(j+1)T]$. Otherwise, condition 2 holds, which implies that $(q(jT),x(jT)) \in S_{N-j}^H \setminus S_0^H$. Let $k_0 = k_{\min}(q(jT),x(jT))$. From the definition of $k_{\min}$, we can infer that $0 < k_0 \leq N-j$ and that $(q(jT),x(jT)) \in S_{k_0}^H \setminus S_{k_0-1}^H$. Thus, $(q(jT),x(jT)) \in \mathscr{RA}_T^H(S_{k_0-1}^H, A^H)$, and for any choice of control $(\sigma_j,u_j) \in F^{RA}(q(jT),x(jT))$, the resulting one step trajectory satisfies $(q((j+1)T),x((j+1)T)) \in S_{k_0-1}^H$ and $(q(t),x(t)) \notin A^H$, $\forall t \in [jT,(j+1)T]$. By the assumption on the control policy $\tilde{\mu}$ and the observation that $S_{k_0-1}^H \subseteq S_{N-j-1}^H$, it then follows that condition 2 holds on $[0,(j+1)T]$. The result then follows by induction. □

Similarly as in the safety problem, we can compute the collections of level set functions representing S_j^H using Algorithm 3.5.1. These functions can be then stored as lookup tables for the online extraction of control inputs, for example, according to Algorithm 3.5.2.

Similarly as in the implementation of the safe control policy, checking the set-membership conditions in Algorithm 3.5.2 is equivalent to checking inequality conditions with respect to level set representations of the sets S_j^H and $\mathscr{RA}_T^{q',\tilde{u}}(S_{j-1}^H(q'),A^H(q'))$.

Algorithm 3.5.2 Online Implementation of Finite Horizon Reach-avoid Control Policy

Require: S_j^H, $j = 1,...,N$, $(q(0),x(0)) \in S_N^H$

for $k = 0$ to $N-1$ **do**
  Measure state $(q(kT),x(kT))$
  if $(q(kT),x(kT)) \in R^H \setminus A^H$ **then**
    Terminate algorithm;
  else
    $F^{RA} \Leftarrow \emptyset$
    Find minimum j such that $(q(kT),x(kT)) \in S_j^H$;
    for all $(\sigma,\tilde{u}) \in \Sigma \times \tilde{U}$ **do**
      $q' \Leftarrow \delta(q(kT),\sigma)$;
      if $x(kT) \in \mathscr{R}\mathscr{A}_T^{q',\tilde{u}}(S_{j-1}^H(q'),A^H(q'))$ **then**
        Add $(\sigma,\tilde{u})$ to F^{RA};
      end if
    end for
    Apply input $(\sigma_k,\tilde{u}_k) \in F^{RA}$;
  end if
end for

3.6 Experimental Results

In this section, we will discuss an experimental application of the proposed controller synthesis algorithms to a quadrotor helicopter platform – the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC). For a comprehensive overview of the development of this platform and its aerodynamic modeling, the interested reader may refer to Hoffmann et al. (2007). In our experiments, a hover control problem is considered, with the objective of controlling a quadrotor helicopter to reach a hover region over a stationary or moving ground target, while satisfying a velocity constraint, and then remain within the hover region, regardless of possible movements by the ground target (see Figure 3.4).

Given a previously designed inner attitude control loop, the hover control problem involves the selection of pitch and roll angles in order to effect changes in the position and velocity of the quadrotor. In particular, the pitch and roll commands are selected from a discrete set, thus resulting in a switching control problem. Under these commands, the relative dynamics between the quadrotor and the ground target can be modeled as follows.

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{y_1} \\ \dot{y_2} \end{bmatrix} = \begin{bmatrix} x_2 + d_1 \\ g\sin(\phi) + d_2 \\ y_2 + d_3 \\ g\sin(-\theta) + d_4 \end{bmatrix} \tag{3.17}$$

Here, the state variables x_1, x_2 and y_1, y_2 denote the relative position and velocity between the quadrotor and the ground target, in the x and y directions, respectively; ϕ and θ are the roll and

Figure 3.4: Setup of hover control experiments. Here the ground target is a radio-controlled car.

pitch angles of the quadrotor; g denotes the gravitational constant; $d = (d_1, d_2, d_3, d_4)$ is a set of disturbance parameters. In particular, d_1 and d_3 are used to capture the effects of unmodelled dynamics, while d_2 and d_4 represent motor noise and also the acceleration of the ground target.

In the experiment scenario, the hover region can be encoded as a set $W^H \subset \mathbb{R}^4$ centered around the origin in the relative position-velocity space. Similarly, the velocity constraint can be encoded as an avoid region $A^H \subset \mathbb{R}^4$. A brief summary of the problem parameters is given below.

- Hover Region (W^H): $|x_1|, |y_1| \leq 0.3$ m, $|x_2|, |y_2| \leq 0.5$ m/s
- Avoid Region (A^H): $|x_2|, |y_2| > 1$ m/s
- Time Step (T): 0.1 seconds
- Time Horizon for Reaching W^H (N): 25 time steps
- Range of Attitude Commands (ϕ, θ): -10, -7.5, -5, -2.5, 0, 2.5, 5, 7.5, 10 degrees
- Disturbance Bounds: $|d_1|, |d_3| \leq 0.1$ m/s, $|d_2|, |d_4| \leq 0.5$ m/s^2

It is important to note that the choice of disturbance bounds is a trade-off between the level of robustness and the feasibility of the control problem. Although a larger disturbance bound may account for a wider range of uncertainties, the feasible set for the controller would in general also be smaller (possibly empty if the bound is sufficiently large). The bounds given here for d_1, d_3 represent about $\pm 10\%$ of the maximum allowed velocity, while the bounds on d_2, d_4 represent about $\pm 30\%$ of the maximum allowed acceleration.

We observe that the hover control problem as defined above is a particular instantiation of Problem 2.2 given in section 2.4.2 for a semiautomated sequential transition system. In particular,

the problem can be separated into two stages. During the first stage, the objective is to reach the hover region W^H in finite time while avoiding A^H (a reach-avoid problem). During the second stage, the objective is to remain inside the hover region W^H (an invariance problem). Thus, we can employ a design procedure that is specialized from the one given in section 2.6.2.

1. Use the method in section 3.4.3 to compute an infinite horizon safe set G^{∞}_{Safe} with respect to W^H and an infinite horizon safe control policy F^{Safe} with respect to G^{∞}_{Safe}.

2. If $G^{\infty}_{Safe} \neq \emptyset$, choose a target set $R^H \subseteq G^{\infty}_{Safe}$.

3. Use the method in section 3.5 to compute a horizon-N reach-avoid set G^N_{RA} with respect to R^H and A^H, as well as a horizon-N reach-avoid control policy F^{RA} with respect to G^N_{RA}.

4. Choose a switching policy as follows. In stage 1, select control inputs according to F^{RA} until $(q(kT), x(kT)) \in R^H$ for some $k \in \{0, 1, ..., N\}$, then switch to F^{Safe}.

Following this procedure, we first compute an infinite horizon safe set $G^{\infty}_{Safe} \subset W^H$ for which the hover objective is feasible. This set is plotted in Figure 3.5. For the finite horizon reach-avoid problem, we select the target set as $R^H = \{(x_1, x_2, y_1, y_2) : |x_1|, |y_1| \leq 0.2\text{m}, |x_2|, |y_2| \leq 0.2\text{m/s}\} \subset G^{\infty}_{Safe}$.

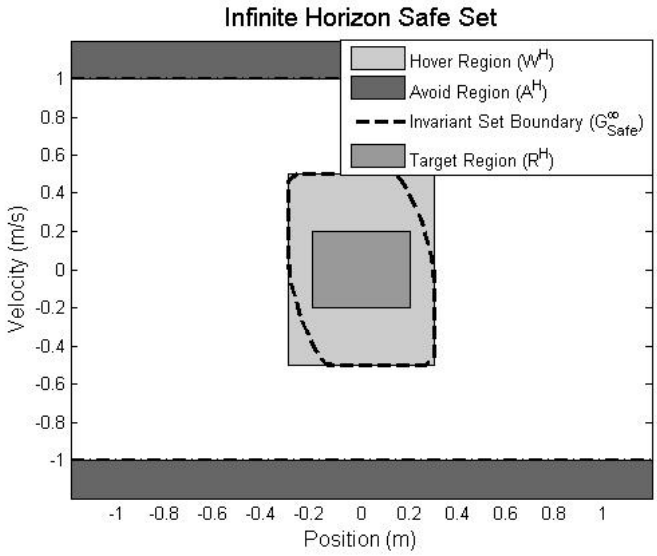
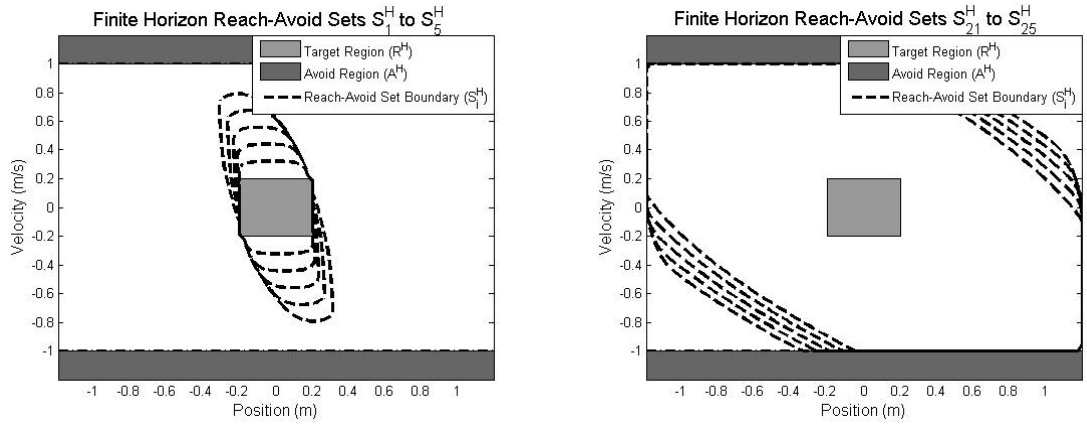


Figure 3.5: Infinite horizon safe set (dashed line) computed for hover objective. Inner rectangle is the target region chosen for reach-avoid problem.

Using the procedures given in section 3.5, we then compute the set of initial conditions which can reach the target set R^H within the time horizon of interest, while satisfying the velocity constraint A^H. Some examples of the sets S^H_j, $j = 1, 2, ..., 25$, as generated by Algorithm 3.5.1, are plotted in Figure 3.6.



(a) (b)

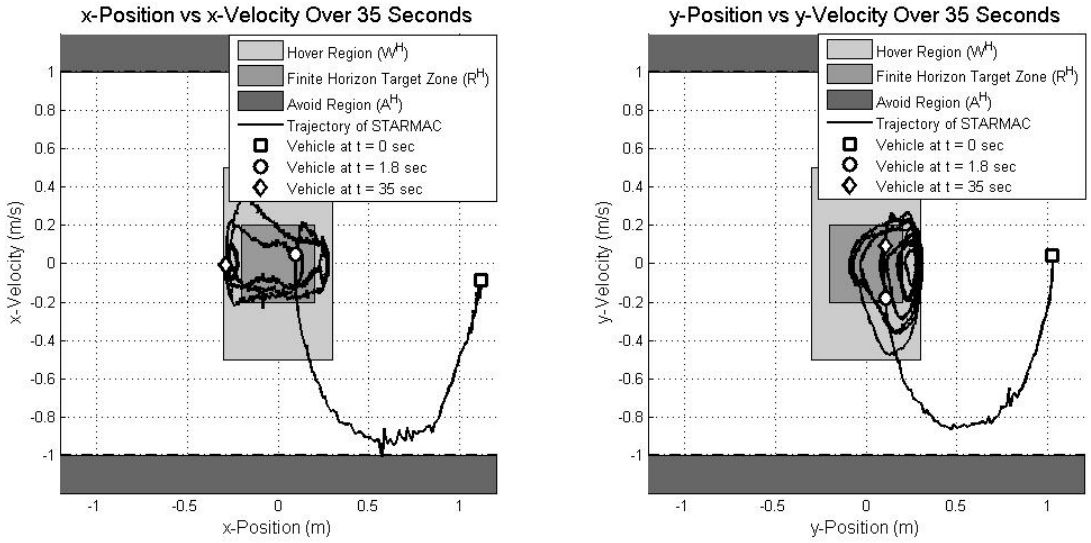
Figure 3.6: Finite horizon reach-avoid sets (dashed lines): (a) sets S_1^H (inner-most line) through S_{10}^H (outer-most line); (b) sets S_{21}^H (inner-most line) through S_{25}^H (outer-most line).

The level set representations of the safe sets and reach-avoid sets, as computed in an offline setting, are stored in lookup table form in the on-board computer. During the experiments, the online selection of control inputs are carried out through an implementation of Algorithms 3.4.2 and 3.5.2. In particular, we obtain sampled measurements of the quadrotor and ground target positions from a VICON camera system. The position measurements are used to estimate the velocity through a first order finite difference scheme. These state values are then used to compute the relative states and select the appropriate pitch and roll commands by checking containment of the current state in particular safe sets or reach-avoid sets. As discussed in sections 3.4.2 and 3.5.2, this check can be performed by checking inequalities with respect to level set representations of stored reachable sets. Given that the VICON system resolves positions to the order of 10^{-3}m, the assumption on precise state measurements is reasonably accurate for these experiments.

The results of an experimental trial in which the ground target is stationary is shown in Figure 3.7. Here the quadrotor is initialized at a state $(x_1, x_2, y_1, y_2) = (1, 0, 1.1, 0)$ m, inside the reach-avoid set S_{25}^H, with the ground target placed at the origin. In the first stage of the experiment, the reach-avoid controller is shown to drive the system trajectory inside the target set R^H within about 1.8 seconds (the allowed time horizon is 2.5 seconds), without exceeding the admissible velocity bounds of ± 1 m/s. For the second stage of the experiment, the safety controller is shown to keep the system state within the hover region W^H for almost the entire remaining 33 seconds of the experiment, except for a brief violation of about 0.2 seconds in duration. The violation can be in part attributed to an observed lag in system response under attitude control commands, which can be accounted for either through a higher order system model or by enlarging the disturbance bound estimates reported here.

From a plot of the attitude commands issued during the first 5 seconds of this experiment (see Figure 3.8), it can be observed that the reach-avoid controller has the characteristics of a

minimum-time-to-reach controller. Namely, an aggressive acceleration action is applied until the state trajectory approaches the velocity constraint, after which an aggressive deceleration action is applied as the quadrotor nears the origin. On the other hand, during the hover phase, the safety controller has the characteristics of a least restrictive controller. Namely, it intervenes only when there is a possibility that the state trajectory will exit the hover region.



(a) (b)

Figure 3.7: Results from hover control experiment over 35 Seconds: (a) x-position (m) and x-velocity (m/s) trajectory of STARMAC; (b) y-position (m) and y-velocity (m/s) trajectory of STARMAC.

The trajectory plot from an experimental trial with a moving ground vehicle is shown in Figure 3.9. In this case, the quadrotor first reaches the hover region over the ground vehicle within 2.1 seconds and then proceeds to track the unplanned movements of the ground vehicle over the course of approximately 44 seconds. The results show that the quadrotor vehicle indeed remains within the hover region, except for two brief violations due to occasional bursts of acceleration by the ground vehicle not accounted for in the disturbance bound estimates. As in the previous experiment, the hover region is quickly recovered within 0.1s and 0.6s, respectively, using the reach-avoid control law (3.16).

3.7 Application to Sequential Reachability Problems

In order to illustrate how the controller synthesis procedure described in this chapter can be applied to problems with sequential reachability objectives, we will revisit in this section the example of automated aerial refueling (AAR) as introduced in Section 2.8. In particular, we consider the

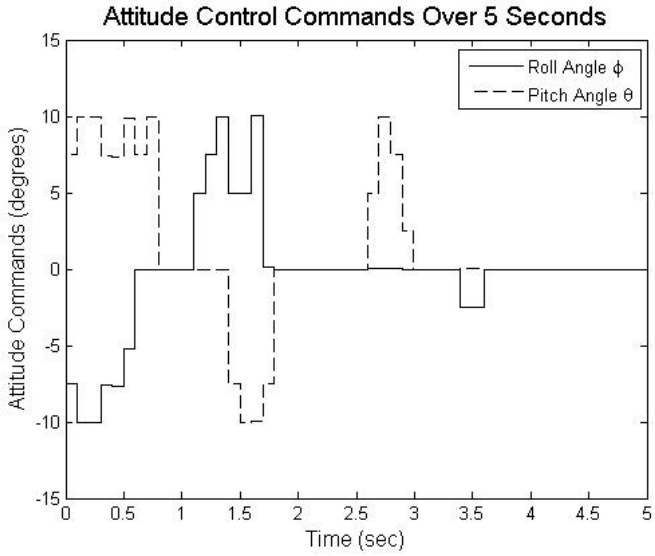
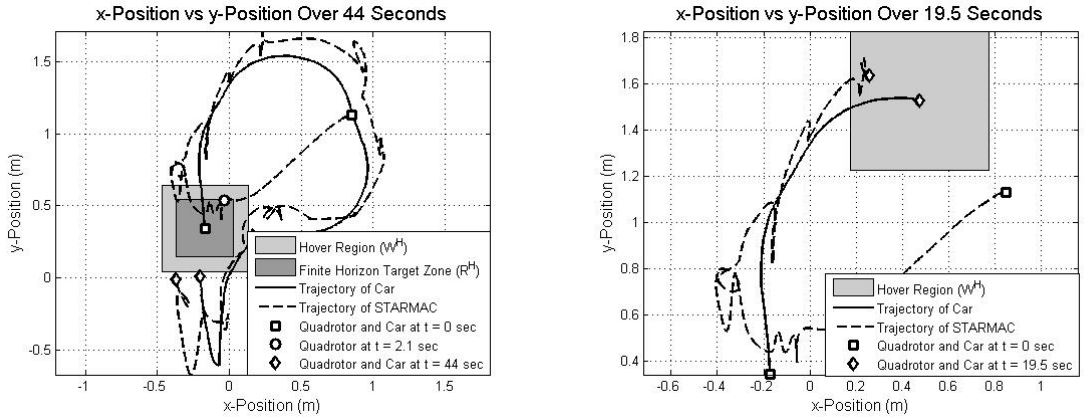


Figure 3.8: Control input plots for hover control experiment over 5 second interval.



(a) (b)

Figure 3.9: Results from car following experiment: (a) x-Position (m) and y-Position (m) trajectories of STARMAC and ground vehicle over 44 second; (b) Snapshot of trajectories at $t = 19.5$.

problem of synthesizing the control law for each maneuver in the refueling sequence as a switching control policy between a finite set of flight modes.

More specifically, we assume the system dynamics and the specifications of target sets and avoid sets as given in Section 2.8 of the preceding chapter. For each maneuver in the refueling sequence, we will use the two-mode flight control system from section 3.2 for the synthesis of UAV controls (see Figure 3.1) to satisfy the reach-avoid objective. In the straight mode, the linear velocity bounds are given by $[\underline{u}_1, \overline{u}_1] = [40, 113]$ m/s, with three quantization levels; while in

the turn mode, the angular velocity bounds are given by $[\underline{u}_2, \overline{u}_2] = [-\pi/6, \pi/6]$ rad/s, with two quantization levels.

Now consider the sequential reachability problem with reach-avoid objectives (Problem 2.1), within the context of AAR. The control design can be carried out as a specialization of the procedure outlined in section 2.6.1, starting with the Rejoin maneuver ($j = 6$).

1. Use the method in section 3.5 to compute a reach-avoid set $G_{RA}^{N_j}$ with respect to R_j and A_j, until the first integer N_j such that $R_{j-1} \subset G_{RA}^{N_j}$.

2. Use equation (3.16) to synthesize a reach-avoid control policy F_j^{RA} with respect to $G_{RA}^{N_j}$.

3. Repeat the above steps for maneuver $j-1$ until the Detach 1 maneuver ($j = 1$). For Detach 1, set $R_0 = X_0$.

4. Choose a switching policy as follows. In maneuvers $j = 0, 1, ..., 5$, select control inputs according to F_j^{RA} until $(q(kT), x(kT)) \in R_j$ for some $k \in \{0, 1, ..., N_j\}$, then reset k to zero and switch to F_{j+1}^{RA}.

A reach-avoid set calculation is performed using Algorithm 3.5.1, with sampling interval $T = 0.1$ seconds, for each of the refueling maneuvers. The reach-avoid set for the *Contact* maneuver is shown in Figure 3.10, computed over a time horizon of 2.1 seconds. Note that for convenience, we translated the target set to the origin in these computations.

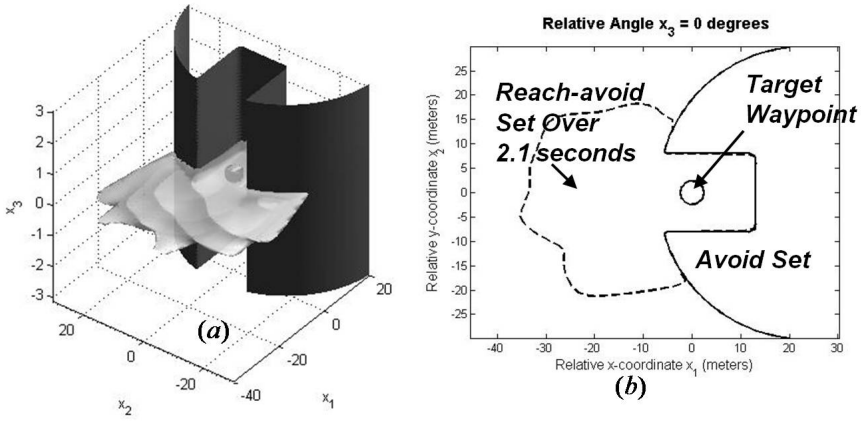


Figure 3.10: Finite horizon reach-avoid set for Contact maneuver: (a) surface plot in relative coordinate space; (b) cross-section at relative angle $x_3 = 0$ degrees.

To validate the resulting reach-avoid sets, a trajectory simulation is performed of the entire aerial refueling sequence. At the beginning of each sampling interval, the UAV receives a state measurement of the relative position and velocity and selects a control input according to the feedback law (3.16). The tanker aircraft then selects a random velocity input from within its input

bounds. The system dynamics is then integrated forward in time according to equation (2.8). The simulation results are given in Figure 3.11, and the plot of the state trajectory in the relative coordinate space is shown in Figure 3.12.

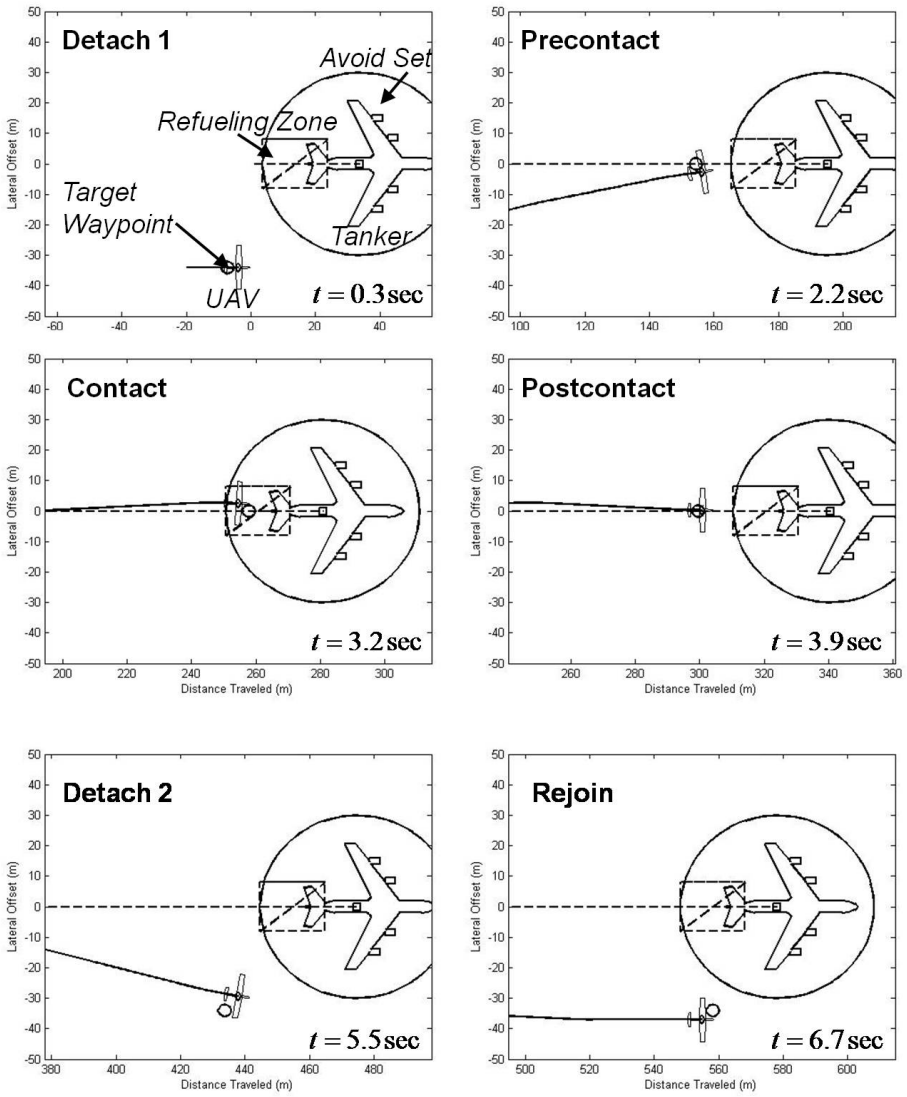


Figure 3.11: Automated aerial refueling sequence simulation sample run.

As can be seen, the UAV successfully avoids a collision with the tanker aircraft, regardless of the random fluctuations of tanker velocity, and completes the entire refueling sequence (excluding the time spent refueling) within 7 seconds. Although not investigated here, extensions to invariance objectives can be carried out by designing infinite horizon safety controllers with respect to the target neighborhoods W_j, and ensuring compatibility between the stationary and transition modes through a specialization of the procedure given in section 2.6.2.

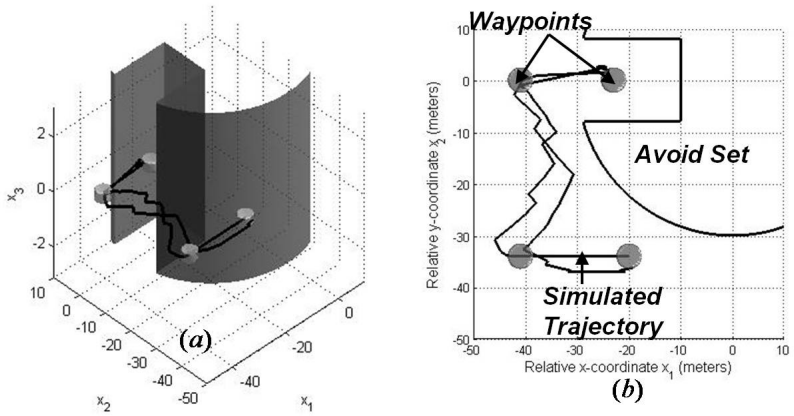


Figure 3.12: Refueling sequence trajectory simulation in relative coordinate space: (a) side view; (b) top-down view.

Part II

Discrete Time Stochastic Hybrid Systems

Chapter 4

Stochastic Game Formulation of Probabilistic Reachability

4.1 Overview and Related Work

In the second part of the dissertation, we will shift our focus to probabilistic reachability problems for stochastic hybrid systems (SHS). The primary difference between a stochastic hybrid system as compared with a deterministic hybrid automaton lies in the model of uncertainty. In the case of a deterministic hybrid system model, such as the one discussed in section 2.3.1, uncertainty in system dynamics is captured through the notion of sets. In particular, the set of admissible disturbance inputs as specified by the disturbance input spaces, along with the set of admissible discrete transitions as specified by the reset relation, implicitly define a set of admissible system trajectories under a particular choice of initial condition and control input. In the case of a stochastic hybrid system model, such as proposed in Altman and Gaitsgory (1997); Hu et al. (2000); Bujorianu and Lygeros (2004); Amin et al. (2006), uncertainty in state evolution is captured through the notion of probability distributions. Speaking somewhat informally, through the introduction of transition probabilities, stochastic differential/difference equations, or transition rates, one implicitly defines a probability distribution over the set of possible executions of a hybrid system.

From a theoretical standpoint, probabilistic models can be viewed as a generalization of deterministic models. Namely, the support of a probability distribution can be interpreted as the set of possible outcomes, while the distribution itself can be interpreted as a quantitative measure of the likelihood of possible outcomes. It is then tempting to conclude that deterministic systems can be studied as special cases of stochastic systems. In practice, however, analysis and computational tools developed for stochastic systems often does not specialize directly to deterministic systems, due to the fact that deterministic dynamics result in degenerate transition probabilities (i.e. probability distributions with mass concentrated at a single point). Furthermore, even if it were possible in certain instances to adapt stochastic techniques to deterministic systems, the process of doing so may overcomplicate the analysis and obscure the intuition behind deterministic problems. Thus, the methods that will be presented for addressing probabilistic reachability problems should not be

interpreted as a generalization of methods addressing deterministic reachability problems. Rather, they should be viewed as an adaptation of reachability analysis to different application scenarios.

Within an application context, there are several possible reasons for employing a probabilistic model as compared with a deterministic model. For cases in which the possible variations in system behavior is not known *a priori* or cannot be conservatively estimated, it may be necessary to model the uncertainties through statistical analysis of empirical data, collected from multiple runs of the system. The resulting model would then have a probabilistic interpretation, namely the probability distribution over the set of possible executions is a quantitative measure of the likelihood that subsets of trajectory, referred to as events, will occur. There are also cases in which the disturbances affecting system dynamics are known to fluctuate within a large range (e.g. wind effects on aircraft trajectory). For such cases, if one were to design controllers with respect to the worst-case realizations of the disturbances, the resulting controllers may be overly conservative. To reduce the conservatism, one may consider adopting a probabilistic disturbance model, which provides an estimate for the likelihood of possible disturbance events. Controller design can be then carried out with respect to probabilistic measures of performance. Finally, if one's objective is to model the aggregate behavior of a large scale system, for example the behavior of economic indices, then a natural modeling framework would be that of a stochastic system. In such cases, the quantitative values of the state variables are generated by the outcomes of a large number of concurrent dynamic processes (e.g. economic activities). As a complete deterministic description of such processes would be intractable for analysis and decision making, a statistical model is often employed instead. In particular, when a variable of interest corresponds to a sum or an average of quantities generated by the underlying processes, then a Gaussian model is a reasonable first order approximation by the Central Limit Theorem.

In the hybrid systems literature, stochastic models have been proposed for application scenarios ranging from air traffic management (Glover and Lygeros, 2004), communication networks (Hespanha, 2004), to systems biology (Hu et al., 2004). For a controlled SHS, the performance of the closed-loop system can be measured in terms of the probability that the system trajectory obeys certain desired specifications. Of interest to safety-critical applications are probabilistic safety and reachability problems in which the control objective is to maximize the probability of remaining within a certain safe set or of reaching a desired target set. In the continuous-time case, a theoretical upper bound on the reachability probability is derived in Bujorianu (2004) using Dirichlet forms. The temporal evolution of the probability density function of the hybrid state has been characterized through generalized Fokker-Planck equations (Bect et al., 2006). Optimal control of stochastic hybrid systems is considered in Bensoussan and Menaldi (2000) and quasi-variational inequalities based on dynamic programming are derived for the optimal trajectory. An optimal control approach towards reachability analysis is discussed in Koutsoukos and Riley (2006) and Mohajerin Esfahani et al. (2011), in which the solutions of probabilistic safety and reachability problems are derived in terms of the viscosity solutions of appropriate Hamiltion-Jacobi-Bellman equations. To address the computational issues associated with probabilistic reachability analysis, the authors in Hu et al. (2005) propose a Markov chain approximation of the SHS using methods from Kushner and Dupuis (1992), while in Prajna et al. (2007), the authors discuss an approach for computing an upper bound on the safety probability using barrier certificates. For discrete-time

stochastic hybrid systems (DTSHS), a theoretical framework for the study of probabilistic safety problems is established in Abate et al. (2008). These results are generalized in Summers and Lygeros (2010) to address the reach-avoid problem, in which the control objective is to reach a desired target set, while remaining within a safe set. Considerations for time-varying and stochastic sets are discussed in Abate et al. (2006) and Summers et al. (2011) respectively.

Recently, we extended the probabilistic safety and reachability of DTSHS, as studied in Abate et al. (2008) and Summers and Lygeros (2010), to a zero-sum stochastic game setting (Kamgarpour et al., 2011). In particular, we considered a scenario in which the evolution of the system state is affected not only by the actions of the control (as in previous work), but also by the actions of a rational adversary, whose objectives are opposed to that of the control. This is motivated by practical applications such as conflict resolution in air traffic management (Tomlin et al., 2002) and control of networked systems subject to external attacks (Amin et al., 2009), in which the intent of certain rational agents may be uncertain. In addition, the framework is applicable to robust control applications, in which there may be unmodeled dynamics whose probability distribution is not known *a priori*. For such cases, a dynamic programming result was stated, without proof, for determining the maximal probability of satisfying the reach-avoid objective, subject to the worst-case adversary behavior, referred to as the *max-min reach-avoid probability*.

The discussions of this chapter is a significant expansion upon the basic problem formulation and the statement of the dynamic programming result given in Kamgarpour et al. (2011). In terms of problem formulation, a formal interpretation is given for the max-min probability as the value of a zero-sum Stackelberg stochastic game with the control as the leader. We then provide a detailed proof of the dynamic programming result for the existence and computation of this value. In the process of the proof, sufficient conditions of optimality are derived for both the control and the adversary. Furthermore, it is briefly discussed how this result, shown for the case of the reach-avoid problem, can be specialized to address the safety problem. For applications with less conservative assumptions on the disturbance, we also investigate the implications of considering alternative information patterns in the problem formulation. In particular, it is shown that the existence of value under symmetric information patterns in general requires randomized player policies. Finally, we discuss in detail the infinite horizon properties of the dynamic programming algorithms, and provide some results on the computation of the infinite horizon value and the existence of infinite horizon optimal policies.

In comparison with existing results in literature, our main contributions are summarized as follows. First, by introducing adversarial inputs into the system model, we formulate a modeling framework which allows analysis of hybrid systems with both stochastic and bounded uncertainties. Second, through our dynamic programming result, we establish a basis for computational algorithms addressing probabilistic safety and reachability problems posed under this modeling framework. Third, the proof of our main result presents a generalization of the stochastic optimal control arguments employed in Abate et al. (2008) and Summers and Lygeros (2010) for single-player probabilistic safety and reachability problems. In particular, measurability properties, which are vital for ensuring that the probabilities of interest can be computed by a recursive procedure, are more difficult to establish in a stochastic game setting as compared with a single-player setting (Nowak, 1985). Thus, our dynamic programming arguments require the use of results from

the analysis of zero-sum stochastic games (Shapley, 1953; Maitra and Parthasarathy, 1970; Kumar and Shiau, 1981; Nowak, 1985; Rieder, 1991; Maitra and Sudderth, 1998; Gonzalez-Trejo et al., 2002), with adjustments to account for the sum-multiplicative form of our utility function and the asymmetric information pattern in a max-min control problem.

This chapter is organized as follows. In section 4.2, we discuss the model for a discrete-time stochastic hybrid game (DTSHG). In section 4.3, we give a formal stochastic game formulation of the probabilistic reach-avoid problem. In section 4.4, we state and prove our main result for computing the max-min reach-avoid probability, and give sufficient conditions of optimality for both the control and the adversary. This is followed by the specialization of this result to the safety problem. In section 4.5, we consider the implications of alternative information patterns on the existence of value and optimal policies. In section 4.6, we discuss the extension of the results to infinite horizon reachability problems. In section 4.7, the proposed methodology is applied to stochastic formulations of the target tracking and aircraft conflict resolution problems as considered in chapter 3. The examples are used to illustrate the utility of stochastic models, the computation of max-min probabilities and control policies, and the interpretation of the dynamic programming results within an application context.

4.2 Discrete-Time Stochastic Hybrid Game Model

The model for a discrete-time stochastic hybrid game (DTSHG) as described in this section is an extension of the discrete-time stochastic hybrid systems (DTSHS) model proposed in Abate et al. (2008); Summers and Lygeros (2010) to a two-player stochastic game setting. As in previous work, we require the stochastic transition kernels to be Borel-measurable and denote by $\mathscr{B}(\cdot)$ the Borel $\sigma-$algebra. This condition ensures that the probabilities of interest can be computed by integration of the transition kernels over a hybrid state space. Following standard conventions in two-player games, we refer to the control as player I and the adversary as player II.

Definition 4.1 (DTSHG)**.** A discrete-time stochastic hybrid game between two players is a tuple $\mathscr{H} = (Q, n, C_a, C_b, \nu_x, \nu_q, \nu_r)$, defined as follows.

- *Discrete state space* $Q := \{q_1, q_2, ..., q_m\}$, $m \in \mathbb{N}$;
- *Dimension of continuous state space* $n : Q \to \mathbb{N}$: a map which assigns to each discrete state $q \in Q$ the dimension of the continuous state space. The hybrid state space is given by $S := \bigcup_{q \in Q} \{q\} \times \mathbb{R}^{n(q)}$;
- *Player I controls* C_a: a nonempty, compact Borel space;
- *Player II controls* C_b: a nonempty, compact Borel space;
- *Continuous state transition kernel* $\nu_x : \mathscr{B}(\mathbb{R}^{n(\cdot)}) \times S \times C_a \times C_b \to [0,1]$: a Borel-measurable stochastic kernel on $\mathbb{R}^{n(\cdot)}$ given $S \times C_a \times C_b$ which assigns to each $s = (q,x) \in S$, $a \in C_a$ and $b \in C_b$ a probability measure $\nu_x(\cdot|s,a,b)$ on the Borel space $(\mathbb{R}^{n(q)}, \mathscr{B}(\mathbb{R}^{n(q)}))$;

- *Discrete state transition kernel* $\nu_q : Q \times S \times C_a \times C_b \to [0,1]$: a Borel-measurable discrete stochastic kernel on Q given $S \times C_a \times C_b$ which assigns to each $s \in S$ and $a \in C_a$, $b \in C_b$ a probability distribution $\nu_q(\cdot|s,a,b)$ over Q;

- *Reset transition kernel* $\nu_r : \mathscr{B}(\mathbb{R}^{n(\cdot)}) \times S \times C_a \times C_b \times Q \to [0,1]$: a Borel-measurable stochastic kernel on $\mathbb{R}^{n(\cdot)}$ given $S \times C_a \times C_b \times Q$ which assigns to each $s \in S$, $a \in C_a$, $b \in C_b$ and $q' \in Q$ a probability measure $\nu_r(\cdot|s,a,b,q')$ on the Borel space $(\mathbb{R}^{n(q')}, \mathscr{B}(\mathbb{R}^{n(q')}))$.

In contrast with the single-player case, the stochastic transition kernels in a DTSHG are affected by the inputs of two agents with possibly differing objectives. In particular, we assume that player I and player II are non-cooperative and consider a conservative decision model in which the actions of player II may be chosen in a rational fashion based upon the actions of player I.

Definition 4.2. A Markov policy for player I is a sequence $\mu = (\mu_0, \mu_1, ..., \mu_{N-1})$ of Borel measurable maps $\mu_k : S \to C_a$, $k = 0, 1, ..., N-1$. The set of all admissible Markov policies for player I is denoted by $\mathscr{M}_a$.

Definition 4.3. A Markov strategy for player II is a sequence $\gamma = (\gamma_0, \gamma_1, ..., \gamma_{N-1})$ of Borel measurable maps $\gamma_k : S \times C_a \to C_b$, $k = 0, 1, ..., N-1$. The set of all admissible Markov strategies for player II is denoted by Γ_b.

The scenario described here is a common setting in robust control problems in which the control selects inputs in anticipation of the worst-case response by an adversary or a disturbance. More formally, this can be interpreted as a zero-sum Stackelberg game in which player I is the leader. Due to the asymmetry in information in a Stackelberg game, equilibrium strategies of a zero-sum game can be typically chosen to be deterministic rather than randomized (Breton et al., 1988). We note, however, that in a zero-sum stochastic game with symmetric information (the actions of player I are not revealed to player II), the existence of a non-cooperative equilibrium in general requires randomized strategies (see for example Shapley, 1953; Maitra and Parthasarathy, 1970). This case is discussed in section 4.5. Furthermore, if one were to consider transition probabilities and utility functions which depend on the entire history of the game, it may also be necessary to broaden the class of player strategies to encompass non-Markov policies (Rieder, 1991; Maitra and Sudderth, 1998). However, as shown in Rieder (1991), when the time horizon is finite, the transition probabilities are Markovian, and the utility function is sum-multiplicative, it is sufficient to consider the class of Markov control policies. The consideration of infinite horizon problems, on the other hand, in general requires semi-Markov control policies, which depend on the initial condition. This case is discussed in section 4.6.

For a given initial condition $s(0) = (q_0, x_0) \in S$, player I policy $\mu \in \mathscr{M}_a$, and player II strategy $\gamma \in \Gamma_b$, the semantics of a DTSHG can be described as follows. At time step k, each player obtains a measurement of the current system state $s(k) = (q(0), x(0)) \in S$. Using this information, player I selects a control input $a(k) = \mu_k(s(k))$, following which player II selects a disturbance input $b(k) = \gamma_k(s(k), a(k))$. The discrete state is then updated according to the discrete transition kernel as $q(k+1) \sim \nu_q(\cdot|s(k), a(k), b(k))$. If the discrete state remains the same, namely $q(k+1) = q(k)$,

then the continuous state is updated according to the continuous state transition kernel as $x(k+1) \sim \nu_x(\cdot|s(k),a(k),b(k))$. On the other hand, if there is a discrete jump, the continuous state is instead updated according to the reset transition kernel as $x(k+1) \sim \nu_r(\cdot|s(k),a(k),b(k),q(k+1))$.

Following this description, we can compose the transition kernels ν_x, ν_q, and ν_r to form a hybrid state transition kernel $\nu : \mathscr{B}(S) \times S \times C_a \times C_b \to [0,1]$ which describes the evolution of the hybrid state under the influence of player I and player II inputs:

$$\nu\big((q',dx')|(q,x),a,b,q'\big) = \begin{cases} \nu_x(dx'|(q,x),a,b)\nu_q(q|(q,x),a,b), & \text{if } q' = q \\ \nu_r(dx'|(q,x),a,b,q')\nu_q(q'|(q,x),a,b), & \text{if } q' \neq q. \end{cases}$$

Using the transition kernel ν, we can now give a formal definition for the executions of a DTSHG.

Definition 4.4. Let $\mathscr{H}$ be a DTSHG and $N \in \mathbb{N}$ be a finite time horizon. For a given $\mu \in \mathscr{M}_a$, $\gamma \in \Gamma_b$, and $s_0 = (q_0,x_0) \in S$, a stochastic process $\{s(k),\ k = 0,...,N\}$ with values in S is an execution of $\mathscr{H}$ if its sample paths are generated according to Algorithm 4.2.1.

Algorithm 4.2.1 DTSHG Execution

Require: Initial condition $s_0 = (q_0,x_0) \in S$, player I policy $\mu \in \mathscr{M}_a$, player II strategy $\gamma \in \Gamma_b$;
  Set $s(0) = s_0$;
  for $k = 0$ to $N-1$ **do**
    Set $a(k) = \mu_k(s(k))$;
    Set $b(k) = \gamma_k(s(k),a(k))$;
    Extract from S a value s_{k+1} for $s(k+1)$ according to $\nu(\cdot|s(k),a(k),b(k))$;
  end for
  return Sample Path $\{s_k, k = 0,...,N\}$.

By this definition, the execution of a DTSHG is a time inhomogeneous stochastic process on the sample space $\Omega = S^{N+1}$, endowed with the canonical product topology $\mathscr{B}(\Omega) := \prod_{k=1}^{N+1} \mathscr{B}(S)$. The evolution of the closed-loop hybrid state trajectory can be described in terms of the transition kernels $\nu^{\mu_k,\gamma_k}(\cdot|s) := \nu(\cdot|s,\mu_k(s),\gamma_k(s,\mu_k(s)))$, $k = 0,\ldots,N$. By Proposition 7.28 of Bertsekas and Shreve (1978), for a given initial condition $s \in S$, player I policy $\mu \in \mathscr{M}_a$, and player II strategy $\gamma \in \Gamma_b$, these stochastic kernels induce a unique probability measure $P_s^{\mu,\gamma}$ on the sample space Ω:

$$P_s^{\mu,\gamma}(S_0 \times S_1 \times \cdots \times S_N) = \int_{S_0}\int_{S_1}\cdots\int_{S_N} \prod_{k=0}^{N-1} \nu^{\mu_k,\gamma_k}(ds_{k+1}|s_k)\delta_s(ds_0), \tag{4.1}$$

where $S_0, S_1, \ldots, S_N \in \mathscr{B}(S)$ are Borel sets and δ_s denotes the probability measure on S which assigns unit mass to the point $s \in S$.

4.2.1 Example - 2-mode Jump Markov System

Consider a simple jump Markov system with two modes of operation $Q = \{q_1, q_2\}$. The transitions between the discrete modes are modeled probabilistically, with the probability of dwelling in mode

q_i given by p_i, $i = 1,2$. While in mode q_i, a continuous state $x \in \mathbb{R}$ evolves according to a stochastic difference equation $x(k+1) = f_i(x(k), a(k), b(k), w(k))$, defined as follows:

$$f_i(x(k), a(k), b(k), w(k)) = \begin{cases} 2x(k) + a(k) + b(k) + w(k), & i = 1 \\ \frac{1}{2}x(k) + a(k) + b(k) + w(k), & i = 2 \end{cases} \tag{4.2}$$

where a and b are player I and player II inputs, and w is a random variable. It is assumed that the players have identical capabilities, with $a, b \in [-1, 1]$. The noise is modeled by a uniform distribution $w \sim \mathscr{U}[-1, +1]$.

Under the DTSHG modeling framework, the hybrid state space is $S = \{q_1, q_2\} \times \mathbb{R}$, and the players' input spaces are $C_a = C_b = [-1, 1]$. The discrete transition kernel ν_q is derived as $\nu_q(q_j|(q_i, x), a, b) = p_i$ if $q_i = q_j$, and $\nu_q(q_j|(q_i, x), a, b) = 1 - p_i$, otherwise. The continuous transition kernel ν_x can be derived from the continuous state dynamics (4.2) as

$$\nu_x(dx'|(q_1, x), a, b) \sim \mathscr{U}[2x + a + b - 1, 2x + a + b + 1],$$
$$\nu_x(dx'|(q_2, x), a, b) \sim \mathscr{U}[\frac{1}{2}x + a + b - 1, \frac{1}{2}x + a + b + 1].$$

Finally, the reset transition kernel is given by $\nu_r(dx'|(q, x), a, b, q') = \nu_x(dx'|(q, x), a, b)$.

4.3 Problem Formulation

Within the context of a DTSHG model, we consider stochastic game formulations of the probabilistic safety and reach-avoid problems. In particular, it is assumed from a robust control standpoint that the objective of player I is to maximize the probability of achieving a given reachability specification, while the objective of player II is to minimize this probability. Thus, the safety and reach-avoid problems for a DTSHG become zero-sum stochastic games. Moreover, due to the fact that player II is allowed to select inputs in response to the actions of player I, they fall within the class of zero-sum Stackelberg games (Breton et al., 1988; Başar and Olsder, 1999). A more precise description of these problems is given below.

First, consider the probabilistic safety problem. Assume that a Borel set $W \in \mathscr{B}(S)$ is given as a safe set. The probability that the sample path $(s_0, s_1, ..., s_N)$ remains in W under fixed choices of $\mu \in \mathscr{M}_a$ and $\gamma \in \Gamma_b$ is given by

$$p_{s_0}^{\mu,\gamma}(W) := P_{s_0}^{\mu,\gamma}(\{(s_0, ..., s_N) : s_k \in W,\ \forall k \in [0, N]\}) = P_{s_0}^{\mu,\gamma}(W^{N+1}), \tag{4.3}$$

where we use $[0, N]$ as a shorthand for $\{0, 1, ..., N\}$.

By Proposition 7.45 of Bertsekas and Shreve (1978), the safety probability in (4.3) can be computed as

$$p_{s_0}^{\mu,\gamma}(W) = \mathbf{1}_W(s_0) \int_{W^N} \prod_{k=0}^{N-1} \nu^{\mu_k,\gamma_k}(ds_{k+1}|s_k) = E_{s_0}^{\mu,\gamma}\left[\prod_{k=0}^{N} \mathbf{1}_W(s_k)\right], \tag{4.4}$$

where $E_{s_0}^{\mu,\gamma}$ denotes the expectation with respect to the probability measure $P_{s_0}^{\mu,\gamma}$ on the sample space Ω. This is analogous the multiplicative payoff given in Abate et al. (2008) for the single-player safety problem.

Now consider the probabilistic reach-avoid problem. Assume that Borel sets $R, W' \in \mathscr{B}(S)$ are given as target set and safe set, respectively, with $R \subseteq W'$. The probability that the sample path $(s_0, s_1, ..., s_N)$ reaches R while staying inside W under fixed choices of $\mu \in \mathscr{M}_a$ and $\gamma \in \Gamma_b$ is given by

$$\begin{aligned} r_{s_0}^{\mu,\gamma}(R,W') &:= P_{s_0}^{\mu,\gamma}(\{(s_0,...,s_N) : \exists k \in [0,N], (s_k \in R) \wedge (s_j \in W',\ \forall j \in [0,k])\}) \\ &= P_{s_0}^{\mu,\gamma}\left(\bigcup_{k=0}^{N} (W' \setminus R)^k \times R\right) = \sum_{k=0}^{N} P_{s_0}^{\mu,\gamma}((W' \setminus R)^k \times R), \end{aligned} \tag{4.5}$$

where the last equality in (4.5) follows by the fact that the union is disjoint. Again by Proposition 7.45 of Bertsekas and Shreve (1978), this probability can be computed as

$$\begin{aligned} r_{s_0}^{\mu,\gamma}(R,W') &= \mathbf{1}_R(s_0) + \mathbf{1}_{W'\setminus R}(s_0) \int_{S^N} \sum_{k=1}^{N} \prod_{j=1}^{k-1} \mathbf{1}_{W'\setminus R}(s_j)\mathbf{1}_R(s_k) \prod_{k=0}^{N-1} \nu^{\mu_k,\gamma_k}(ds_{k+1}|s_k) \\ &= E_{s_0}^{\mu,\gamma}\left[\sum_{k=0}^{N} \left(\prod_{j=0}^{k-1} \mathbf{1}_{W'\setminus R}(s_j)\right) \mathbf{1}_R(s_k)\right], \end{aligned} \tag{4.6}$$

where $E_{s_0}^{\mu,\gamma}$ denotes the expectation with respect to the probability measure $P_{s_0}^{\mu,\gamma}$ on the sample space Ω. This is analogous to the sum-multiplicative payoff given in Summers and Lygeros (2010) for the single-player reach-avoid problem.

The connection between the safety problem and reach-avoid problem is established by the observation that the hybrid state remains inside a safe set W for all $k = 0, 1, ..., N$ if and only if it does not reach the unsafe set $S \setminus W$ for any $k = 0, 1, ..., N$. Mathematically speaking, for any $\mu \in \mathscr{M}_a$ and $\gamma \in \Gamma_b$,

$$p_{s_0}^{\mu,\gamma}(W) = 1 - r_{s_0}^{\mu,\gamma}(S \setminus W, S). \tag{4.7}$$

Through this relation, the computation of the safety probability can be viewed as a special case of the computation of the reach-avoid probability. As such, we will now focus on the reach-avoid problem, with the understanding that any results for the reach-avoid problem can be specialized to the safety problem via equation (4.7).

In a zero-sum Stackelberg, or equivalently a max-min, formulation of the probabilistic reach-avoid problem, the control selects a choice of feedback control policy $\mu \in \mathscr{M}_a$ to maximize (4.6), in anticipation of the worst-case response by the adversary in the selection of the feedback strategy $\gamma \in \Gamma_b$. More specifically, we define the worst-case reach-avoid probability under a player I policy $\mu \in \mathscr{M}_a$ as

$$r_{s_0}^{\mu}(R,W') = \inf_{\gamma \in \Gamma_b} r_{s_0}^{\mu,\gamma}(R,W'),\ s_0 \in S. \tag{4.8}$$

The Stackelberg or max-min pay-off for player I is then given by

$$r_{s_0}^{*}(R,W') := \sup_{\mu \in \mathscr{M}_a} r_{s_0}^{\mu}(R,W'),\ s_0 \in S. \tag{4.9}$$

The optimal policy of player I and the optimal strategy of player II are interpreted in terms of Stackelberg equilibrium strategies. In particular, by the definitions given in Breton et al. (1988) and Başar and Olsder (1999), a policy $\mu^* \in \mathscr{M}_a$ is a Stackelberg equilibrium policy for player I if it satisfies

$$r_{s_0}^{\mu^*}(R,W') = r_{s_0}^*(R,W'),\ \forall s_0 \in S. \tag{4.10}$$

For a given choice of equilibrium policy $\mu^* \in \mathscr{M}_a$ for player I, a strategy $\gamma^* \in \Gamma_b$ is a Stackelberg equilibrium strategy for player II if it satisfies

$$r_{s_0}^{\mu^*,\gamma^*}(R,W') \leq r_{s_0}^{\mu^*,\gamma}(R,W'),\ \forall s_0 \in S, \gamma \in \Gamma_b, \tag{4.11}$$

Any strategy pair (μ^*,γ^*) satisfying (4.10) and (4.11) is referred to as a Stackelberg solution to (4.8) and (4.9). Relating these notions to the probabilistic reachability problem of interest, we will call the Stackelberg payoff the *max-min reach-avoid probability*, an equilibrium policy for player I a *max-min control policy*, and an equilibrium strategy for player II a *worst-case adversary strategy*.

We can now give a precise statement of the probabilistic reach-avoid problem for a DTSHG.

Problem 4.1. Given a DTSHG $\mathscr{H}$, target set $R \in \mathscr{B}(S)$, and safe set $W' \in \mathscr{B}(S)$ such that $R \subseteq W'$:

(I) Compute the max-min reach-avoid probability $r_{s_0}^*(R,W')$, $\forall s_0 \in S$;

(II) Find a max-min control policy $\mu^* \in \mathscr{M}_a$, whenever it exists;

(III) Find a worst-case adversary strategy $\gamma^* \in \Gamma_b$, whenever it exists.

4.4 Max-min Probability Computation

In this section, we provide a detailed proof of the main result from Kamgarpour et al. (2011), as the solution to Problem 4.1. In particular, it will be shown that under certain regularity assumptions, the max-min probability $r_{s_0}^*(R,W')$ can be computed using an appropriate dynamic programming algorithm, and that there exists a max-min Markov policy $\mu^* \in \mathscr{M}_a$ for player I which achieves this probability under the worst-case player II strategy. Following the proof, we will discuss some practical implications of the theorem and specialize the results to a stochastic game formulation of the probabilistic safety problem. Finally, a concrete example will be provided to illustrate the procedure for computing $r_{s_0}^*(R,W')$, as well as the max-min policy $\mu^* \in \mathscr{M}_a$ for player I and the worst-case strategy $\gamma^* \in \Gamma_b$ for player II.

4.4.1 Main Theorem

For our theoretical derivations, we impose the following regularity assumptions.

Assumption 4.1.

(a) For each $s = (q,x) \in S$ and $E_1 \in \mathscr{B}(\mathbb{R}^{n(q)})$, the function $(a,b) \to \nu_x(E_1|s,a,b)$ is continuous;

(b) For each $s = (q,x) \in S$ and $q' \in Q$, the function $(a,b) \to \nu_q(q'|s,a,b)$ is continuous;

(c) For each $s = (q,x) \in S$, $q' \in Q$, and $E_2 \in \mathscr{B}(\mathbb{R}^{n(q')})$, the function $(a,b) \to \nu_r(E_2|s,a,b,q')$ is continuous.

The need for continuity assumptions on the stochastic kernel commonly arise in the stochastic game literature (see for example Kumar and Shiau, 1981; Maitra and Parthasarathy, 1970; Nowak, 1985; Gonzalez-Trejo et al., 2002), due to the difficulties in ensuring the measurability of value functions under max-min dynamic programming operations. Following the approach in Nowak (1985) and Rieder (1991), we only assume continuity of the stochastic kernels in the actions of Player I and Player II, but not necessarily in the system state. This allows for stochastic hybrid systems in which transition probabilities change abruptly with changes in the system state. Furthermore, if the action spaces C_a and C_b are finite or countable, then the assumptions are satisfied under the discrete topology on C_a and C_b. Also, the assumptions on ν_v and ν_r are satisfied if these kernels admit density functions that are continuous in the player inputs.

For the construction of a dynamic programming solution to Problem 4.1, we define a max-min dynamic programming operator $\mathscr{T}$ which takes as its argument a Borel measurable function $J : X \to [0,1]$ and produces another real-valued function on X:

$$\mathscr{T}(J)(s) := \mathbf{1}_R(s) + \sup_{a \in C_a} \inf_{b \in C_b} \mathbf{1}_{W' \setminus R}(s) H(s,a,b,J), \tag{4.12}$$

$$\text{where } H(s,a,b,J) = \int_S J(s') \nu(ds'|s,a,b).$$

We now proceed to show that the max-min reach-avoid probability, along with the max-min control policy and worst-case disturbance strategy, can be derived from a dynamic programming algorithm using the operator $\mathscr{T}$.

Theorem 4.1. *Let $\mathscr{H}$ be a DTSHG satisfying Assumption 4.1. Let $R, W' \in \mathscr{B}(S)$ be Borel sets such that $R \subseteq W'$. Let the operator $\mathscr{T}$ be defined as in (4.12). Then the composition $\mathscr{T}^N = \mathscr{T} \circ \mathscr{T} \circ \cdots \circ \mathscr{T}$ (N times) is well-defined and*

(a) $r^*_{s_0}(R,W') = \mathscr{T}^N(\mathbf{1}_R)(s_0), \forall s_0 \in S$;

(b) There exists a player I policy $\mu^* \in \mathscr{M}_a$ and a player II strategy $\gamma^* \in \Gamma_b$ satisfying

$$r^{\mu,\gamma^*}_{s_0}(R,W') \leq r^*_{s_0}(R,W') = r^{\mu^*,\gamma^*}_{s_0}(R,W') \leq r^{\mu^*,\gamma}_{s_0}(R,W'), \tag{4.13}$$

$\forall s_0 \in S$, $\mu \in \mathscr{M}_a$, and $\gamma \in \Gamma_b$. In particular, μ^* is a max-min control policy, and γ^* is a worst-case adversary strategy.

(c) Let $J^*_N = \mathbf{1}_R$, $J^*_k = \mathscr{T}^{N-k}(\mathbf{1}_R)$, $k = 0,1,...,N-1$. If $\mu^* \in \mathscr{M}_a$ is a player I policy which satisfies

$$\mu^*_k(s) \in \arg\sup_{a \in C_a} \inf_{b \in C_b} H(s,a,b,J^*_{k+1}), \tag{4.14}$$

$\forall s \in W' \setminus R,\ k = 0,1,\ldots,N-1$, *then* μ^* *is a max-min control policy. If* $\gamma^* \in \Gamma_b$ *is a player II strategy which satisfies*

$$\gamma_k^*(s,a) \in \arg\inf_{b \in C_b} H(s,a,b,J_{k+1}^*), \tag{4.15}$$

$\forall s \in W' \setminus R,\ a \in C_a,\ k = 0,1,\ldots,N-1$, *then* γ^* *is a worst-case adversary strategy.*

First, we will present a recursive procedure for computing the reach-avoid probability $r_{s_0}^{\mu,\gamma}(R,W')$ under fixed choices of player I policy $\mu \in \mathscr{M}_a$ and player II strategy $\gamma \in \Gamma_b$. Consider the payoff functions $J_k^{\mu,\gamma} : S \to [0,1]$, $k = 0,\ldots,N$, defined as

$$\begin{aligned} J_N^{\mu,\gamma}(s_N) &= \mathbf{1}_K(s_N), \\ J_k^{\mu,\gamma}(s_k) &= E_{s_k}^{\mu,\gamma}\left[\mathbf{1}_K(s_k) + \sum_{j=k+1}^{N}\left(\prod_{i=k+1}^{j-1}\mathbf{1}_{W'\setminus R}(s_i)\right)\mathbf{1}_R(s_j)\right],\ k = 0,1,...,N-1. \end{aligned} \tag{4.16}$$

From this definition we can infer that $r_{s_0}^{\mu,\gamma}(R,W') = J_0^{\mu,\gamma}(s_0)$, $\forall s_0 \in S$. Now consider a recursion operator $\mathscr{T}_{f,g}$, parameterized by an one-stage player I policy $f : X \to C_a$ and an one-stage player II strategy $g : S \times C_a \to C_b$:

$$\mathscr{T}_{f,g}(J)(s) = \mathbf{1}_R(s) + \mathbf{1}_{W'\setminus R}(s)H(s,f(s),g(s,f(s)),J),\ s \in S, \tag{4.17}$$

where H is defined in (4.12). It can be checked in a straightforward manner that the operator $\mathscr{T}_{f,g}$ satisfies a monotonicity property: for any Borel measurable functions J, J' from S to $[0,1]$ such that $J \le J'$, $\mathscr{T}_{f,g}(J)(s) \le \mathscr{T}_{f,g}(J')(s), \forall s \in S$. This property will become useful later on in the dynamic programming arguments.

The following result provides a recursive algorithm for computing the functions $J_k^{\mu,\gamma}$.

Lemma 4.1. *Let* $\mu \in \mathscr{M}_a$, $\gamma \in \Gamma_b$. *Then the payoff functions* $J_k^{\mu,\gamma}$, $k = 0,1,...,N$ *satisfies*

$$J_k^{\mu,\gamma}(s) = \mathscr{T}_{\mu_k,\gamma_k}(J_{k+1}^{\mu,\gamma})(s),\ \forall s \in S,\ k = 0,1,...,N-1. \tag{4.18}$$

Proof. For the case of $k = N-1$, $J_N^{\mu,\gamma} = \mathbf{1}_R$ implies that for any $s \in S$,

$$\begin{aligned} J_{N-1}^{\mu,\gamma}(s) &= \mathbf{1}_R(s) + \mathbf{1}_{W'\setminus R}(s)\int_S \mathbf{1}_R(s_N)\nu^{\mu_{N-1},\gamma_{N-1}}(ds_N|s) \\ &= \mathscr{T}_{\mu_{N-1},\gamma_{N-1}}(J_N^{\mu,\gamma}). \end{aligned}$$

For the case of $k < N-1$, the expression for $J_k^{\mu,\gamma}$ in (4.16) implies that for any $s \in S$,

$$\begin{aligned} J_k^{\mu,\gamma}(s) =& \mathbf{1}_R(s) + \mathbf{1}_{W'\setminus R}(s)\int_S \mathbf{1}_R(s_{k+1}) + \mathbf{1}_{W'\setminus R}(s_{k+1}) \\ & \left(\int_{S^{N-k-1}} \sum_{j=k+2}^{N}\prod_{i=k+2}^{j-1}\mathbf{1}_{W'\setminus R}(s_i)\mathbf{1}_R(s_j)\right)\prod_{j=k+1}^{N-1}\nu^{\mu_j,\gamma_j}(ds_{j+1}|s_j)\nu^{\mu_k,\gamma_k}(ds_{k+1}|s) \\ =& \mathbf{1}_R(s) + \mathbf{1}_{W'\setminus R}(s)\int_S J_{k+1}^{\mu,\gamma}(s_{k+1})\nu^{\mu_k,\gamma_k}(ds_{k+1}|s). \end{aligned}$$

It follows from definition of $\mathscr{T}_{f,g}$ that the last expression above is $\mathscr{T}_{\mu_k,\gamma_k}(J_{k+1}^{\mu,\gamma})$, thus concluding the proof. □

Next, we will show that under Assumption 4.1, the operator $\mathscr{T}$ defined in (4.12) preserves suitable measurability properties (thus allowing recursive dynamic programming calculations) and that there exists one-stage player I policy and player II strategy achieving the supremum and infimum in (4.12).

In the following, we state a special case of Corollary 1 given in Brown and Purves (1973). This result allows us to show that the operator $\mathscr{T}$ preserves Borel measurability and that it is sufficient to consider Borel measurable selectors.

Lemma 4.2. *Let X, Y be complete separable metric spaces such that Y is compact, and f be a real-valued Borel measurable function defined on $X \times Y$ such that $f(x,\cdot)$ is lower semicontinuous with respect to the topology on Y. Define $f^* : X \to \mathbb{R} \cup \{\pm\infty\}$ by*

$$f^*(x) = \inf_{y \in Y} f(x,y).$$

(a) The set

$$I = \{x \in X : \text{ for some } y \in Y, f(x,y) = f^*(x)\},$$

is Borel measurable.

(b) For every $\varepsilon > 0$, there exists a Borel measurable function $\phi : X \to Y$, satisfying, for all $x \in X$,

$$\begin{aligned} &f(x,\phi(x)) = f^*(x), \text{ if } x \in I, \\ &f(x,\phi(x)) \leq \begin{cases} f^*(x) + \varepsilon, \text{ if } x \notin I, \ f^*(x) > -\infty, \\ -1/\varepsilon, \text{ if } x \notin I, \ f^*(x) = -\infty. \end{cases} \end{aligned}$$

For the purpose of showing that the supremum and infimum in the expression for $\mathscr{T}$ is achieved, we will also need the following technical result.

Lemma 4.3. *Let f be a bounded real-valued Borel measurable function on a Borel space Y, and t be a Borel measurable transition probability from a Borel space X into Y such that $t(B|\cdot)$ is continuous on X for each $B \in \mathscr{B}(Y)$. Then the function $x \to \int f(y)t(dy|x)$ is continuous on X.*

This was stated as Fact 3.9 in Nowak (1985). Since neither a proof nor relevant references are provided in Nowak (1985), and also given that this result is the primary use for Assumption 4.1, a detailed proof is given in appendix A.

We now prove a selection result for the max-min operator $\mathscr{T}$. For notational conveniences, we denote by $\mathscr{F}$ the set of Borel measurable functions from S to $[0,1]$.

Proposition 4.1. *If Assumption 4.1 holds, then*

(a) $\forall J \in \mathscr{F}$, $\mathscr{T}(J) \in \mathscr{F}$;

(b) For any $J \in \mathscr{F}$, there exists a Borel measurable function $g^* : S \times C_a \to C_b$ such that, for all $(s,a) \in S \times C_a$,

$$g^*(s,a) \in \arg\inf_{b \in C_b} H(s,a,b,J);$$

(c) For any $J \in \mathscr{F}$, there exists a Borel measurable function $f^* : S \to C_a$, such that for all $s \in S$,

$$f^*(s) \in \arg\sup_{a \in C_a} \inf_{b \in C_b} H(s,a,b,J).$$

Proof. Let $J \in \mathscr{F}$. Define a function $F_J : S \times C_a \times C_b \to \mathbb{R}$ as $F_J(s,a,b) = H(s,a,b,J)$. From the definition of H, the range of F_J is contained in $[0,1]$. By the Borel measurability of J and ν, Proposition 7.29 of Bertsekas and Shreve (1978) implies that F_J is Borel measurable. Furthermore, by Assumption 4.1 and Lemma 4.3, $F_J(s,a,b)$ is continuous in a and b, for each $s \in S$. Now consider a function $\tilde{F}_J(s,a) = \inf_{b \in C_b} F_J(s,a,b)$. By the compactness of C_b and continuity of F_J in b, this infimum is achieved for each fixed (s,a) (see for example Rudin, 1976). Thus, applying Lemma 4.2, we have that there exists a Borel measurable function $g^* : S \times C_a \to C_b$ for which part (b) holds. Furthermore, by Proposition 7.32 of Bertsekas and Shreve (1978), $\tilde{F}_J$ is continuous in a. Let $F_J^*(s) = \sup_{a \in C_a} \tilde{F}_J(s,a) = -\inf_{a \in C_a} -\tilde{F}_J(s,a)$. Then, by a repeated application of Lemma 4.2, there exists a Borel measurable function $f^* : S \to C_a$ such that part (c) holds. By the composition of Borel measurable functions, this also implies that F_J^* is Borel measurable.

Finally, it can be observed that $\mathscr{T}(J)(s) = \mathbf{1}_R(s) + \mathbf{1}_{W' \setminus R}(s) F_J^*(s)$, $\forall s \in S$. Given that Borel measurability is preserved under summation and multiplication (see for example Folland, 1999, Proposition 2.6), $\mathscr{T}(J)$ is Borel measurable. It is also clear that $0 \leq \mathscr{T}(J) \leq 1$. Part (a) then follows. □

In the following two propositions, we show by a dynamic programming argument that $\mathscr{T}^N(\mathbf{1}_R)$ both upper bounds and lower bounds the max-min reach-avoid probability.

Proposition 4.2.

(a) $\forall s_0 \in S$, $\mathscr{T}^N(\mathbf{1}_R)(s_0) \leq r^*_{s_0}(R,W')$;

(b) There exists $\mu^* \in \mathscr{M}_a$ such that, for any $\gamma \in \Gamma_b$, $\mathscr{T}^N(\mathbf{1}_R)(s_0) \leq r^{\mu^*,\gamma}_{s_0}(R,W'), \forall s_0 \in S$.

Proof. For notational convenience, we define $J_k^* := \mathscr{T}^{N-k}(\mathbf{1}_R)$, $k = 0,1,...,N$. First, we prove the following claim by backwards induction on k: there exists $\mu^*_{k \to N} = (\mu_k^*, \mu_{k+1}^*, ..., \mu_{N-1}^*) \in \mathscr{M}_a$ such that, for any $\gamma_{k \to N} = (\gamma_k, \gamma_{k+1}, ..., \gamma_{N-1}) \in \Gamma_b$, $J_k^* \leq J_k^{\mu^*_{k \to N}, \gamma_{k \to N}}$.

Let $\gamma_{k \to N} \in \Gamma_b$ be arbitrary. The case of $k = N$ is trivial. Now assume that this holds for $k = h$. Let $\mu^*_{h \to N} \in \mathscr{M}_a$ be a player I policy satisfying the induction hypothesis. By Proposition 4.1(c), there exists a Borel measurable function $f^* : X \to C_a$ such that

$$f^*(s) \in \arg\sup_{a \in C_a} \inf_{b \in C_b} H(s,a,b,J_h), \ \forall s \in S.$$

Choose a policy $\mu^*_{h-1\to N} = (f^*, \mu^*_{h\to N})$. Then by the monotonicity of the operator $\mathscr{T}_{f,g}$ and Lemma 4.1, we have for each $s \in S$:

$$\begin{aligned} J_{h-1}^{\mu^*_{h-1\to N},\gamma_{h-1\to N}}(s) &= \mathscr{T}_{f^*,\gamma_{h-1}}(J_h^{\mu^*_{h\to N},\gamma_{h\to N}})(s) \geq \mathscr{T}_{f^*,\gamma_{h-1}}(J_h^*)(s) \\ &= \mathbf{1}_R(s) + \mathbf{1}_{W'\setminus R}(s) H(s, f^*(s), \gamma_{h-1}(s, f^*(s)), J_h^*) \\ &\geq \mathbf{1}_R(s) + \inf_{b\in C_b} \mathbf{1}_{W'\setminus R}(s) H(s, f^*(s), b, J_h^*) \\ &= \mathscr{T}(J_h^*)(s) = J_{h-1}^*(s). \end{aligned}$$

The claim then follows by induction. From this, we obtain $\mu^*_{0\to N} \in \mathscr{M}_a$ satisfying $\mathscr{T}^N(\mathbf{1}_R)(s_0) = J_0^*(s_0) \leq J_0^{\mu^*_{0\to N},\gamma_{0\to N}}(s_0) = r_{s_0}^{\mu^*_{0\to N},\gamma_{0\to N}}(R,W')$, $\forall s_0 \in S$, $\gamma_{0\to N} \in \Gamma_b$, and hence satisfying statement (b). Furthermore, since $\gamma_{0\to N}$ is arbitrary, $\mathscr{T}^N(\mathbf{1}_R)(s_0) \leq \inf_{\gamma\in\Gamma_b} r_{s_0}^{\mu^*_{0\to N},\gamma}(R,W')$, $\forall s_0 \in S$. Statement (a) then follows. □

Proposition 4.3.

(a) $\forall s_0 \in S$, $r^*_{s_0}(R,W') \leq \mathscr{T}^N(\mathbf{1}_R)(s_0)$;

(b) There exists $\gamma^* \in \Gamma_b$ *such that, for any* $\mu \in \mathscr{M}_a$, $r_{s_0}^{\mu,\gamma^*}(R,W') \leq \mathscr{T}^N(\mathbf{1}_R)(s_0), \forall s_0 \in S$.

Proof. As in the proof of Proposition 4.2, we define $J_k^* := \mathscr{T}^{N-k}(\mathbf{1}_R)$, $k = 0,1,...,N$. First, we prove the following claim by backwards induction on k: there exists $\gamma^*_{k\to N} = (\gamma^*_k, \gamma^*_{k+1}, ..., \gamma^*_{N-1}) \in \Gamma_d$ such that, for any $\mu_{k\to N} = (\mu_k, \mu_{k+1}, ..., \mu_{N-1}) \in \mathscr{M}_a$, $J_k^{\mu,\gamma^*_{k\to N}} \leq J_k^*$.

Let $\mu_{k\to N} \in \mathscr{M}_a$ be arbitrary. The case of $k = N$ is trivial. Now assume that this holds for $k = h$. Let $\gamma^*_{h\to N} \in \Gamma_b$ be a player II strategy satisfying the induction hypothesis. By Proposition 4.1(b), there exists a Borel measurable function $g^* : S \times C_a \to C_b$ such that

$$g^*(s,a) \in \arg\inf_{b\in C_b} H(s,a,b,J_h),\ \forall s \in S, a \in C_a.$$

Choose a strategy $\gamma^*_{h-1\to N} = (g^*, \gamma^*_{h\to N})$. Then we have for each $s \in S$:

$$\begin{aligned} J_{h-1}^{\mu_{h-1\to N},\gamma^*_{h-1\to N}}(s) &= \mathscr{T}_{\mu_{h-1},g^*}(J_h^{\mu_{h\to N},\gamma^*_{h\to N}})(s) \leq \mathscr{T}_{\mu_{h-1},g^*}(J_h^*)(x) \\ &= \mathbf{1}_R(s) + \mathbf{1}_{W'\setminus R}(s) H(s, \mu_{h-1}(s), g^*(s, \mu_{h-1}(s)), J_h^*) \\ &= \mathbf{1}_R(s) + \inf_{b\in C_b} \mathbf{1}_{W'\setminus R}(s) H(s, \mu_{h-1}(s), b, J_h^*) \\ &\leq \mathscr{T}(J_h^*)(s) = J_{h-1}^*(s). \end{aligned}$$

The claim then follows by induction. From this, we obtain $\gamma^*_{0\to N} \in \Gamma_d$ satisfying $r_{s_0}^{\mu,\gamma^*_{0\to N}}(R,W') = J_0^{\mu,\gamma^*_{0\to N}}(s_0) \leq J_0^*(s_0) = \mathscr{T}^N(\mathbf{1}_R)(s_0)$, $\forall s_0 \in S$, $\mu \in \mathscr{M}_a$, and hence statement (b). This in turn implies that $r_{s_0}^{\mu}(R,W') = \inf_{\gamma\in\Gamma_b} r_{s_0}^{\mu,\gamma}(R,W') \leq \mathscr{T}^N(\mathbf{1}_R)(s_0)$, $\forall s_0 \in S$, $\mu \in \mathscr{M}_a$, proving statement (a). □

Combining the results of Proposition 4.2 and 4.2, we can now prove Theorem 4.1.

Proof. Statement (a) of Theorem 4.1 follows directly from Proposition 4.2(a) and 4.3(a). The player I policy μ^* and player II strategy γ^* satisfying statement (b) is provided by Proposition 4.2(b) and 4.3(b), respectively. Finally, it can be inferred from the proof of Proposition 4.2 and 4.3 that any player I policy μ^* and player II strategy γ^* satisfying the conditions in statement (c) is a max-min policy or worst-case strategy, respectively. □

4.4.2 Implications of the Main Theorem

1) Probabilistic reachability computation: By statement (a) of Theorem 4.1, the max-min reach-avoid probability can be computed using a sup-inf dynamic programming procedure. This can be viewed as the counterpart to the HJI equation for discrete time stochastic system. Namely, instead of solving a terminal value PDE backwards in time, the probabilistic reachability computation for a DTSHG involves solving an integro-difference equation backwards in time, as described by the dynamic programming operator $\mathscr{T}$. As in the case of the HJI equation, the solution to the integro-difference equation in general does not have a closed-form expression, and as such requires numerical approximation. However, whereas the approximation of the PDE solution involves approximation of the spatial derivatives and an integral in time, the approximation of the solution to the integro-difference equation requires approximation of spatial integrals. To illustrate this, suppose that we have computed the optimal cost-to-go function $J^*_{k+1} := \mathscr{T}^{N-k-1}(\mathbf{1}_R)$ at time $k+1$, then the optimal cost-to-go function J^*_k at time k is computed as

$$J^*_k(s) = \mathscr{T}(J^*_{k+1})(s) = \mathbf{1}_R(s) + \sup_{a\in C_a} \inf_{b\in C_b} \mathbf{1}_{W'\setminus R}(s) H(s,a,b,J^*_{k+1}),\ \forall s \in S. \tag{4.19}$$

For a fixed $s = (q,x) \in S$, and inputs $a \in C_a$, $b \in C_b$, the explicit form of $H(s,a,b,J^*_{k+1})$ in the above expression is given by

$$\begin{aligned} H((q,x),a,b,J^*_{k+1}) &= \int_S J^*_{k+1}(s')\nu(ds'|(q,x),a,b) \\ &= \nu_q(q|(q,x),a,b) \int_{\mathbb{R}^{n(q)}} J^*_{k+1}(q,x')\nu_x(dx'|(q,x),a,b) + \\ &\quad \sum_{q'\neq q} \nu_q(q'|(q,x),a,b) \int_{\mathbb{R}^{n(q')}} J^*_{k+1}(q',x')\nu_r(dx'|(q,x),a,b,q'). \end{aligned} \tag{4.20}$$

Thus, the approximation of $\mathscr{T}$ is tantamount to the approximation of the integral in (4.20) for discretized values of the hybrid state and player inputs. Assuming that the stochastic kernels ν_q and ν_r are described in terms of probability density functions, this approximation can be performed using Riemann or Lebesgue integrals. For the single player case, a numerical scheme of such type is proposed in Abate et al. (2007) for the computation of the safety probability. In particular, it is shown that piecewise constant approximations of the value function on a grid of the continuous state space converge uniformly to the optimal value function, at a rate that is linear in the grid size parameter. We anticipate that a similar result can be shown for the case of a DTSHG. However,

it can be observed that the computational cost of such an approach scales exponentially with the dimension of the continuous state space, which currently limits the application of our approach to problems with continuous state dimensions of $n \leq 4$. The reduction in computation time is a topic of ongoing research (Esmaeil Zadeh Soudjani and Abate, 2011).

2) Controller synthesis: Equations (4.14) and (4.15) provide us with sufficient conditions for optimality of the players' policies and strategies. In particular, this can be used to synthesize a max-min control policy for player I from the value functions computed through the dynamic programming recursion. To illustrate, suppose that the input ranges C_a and C_b along with the state space S has been appropriately discretized into C_a^d, C_b^d, and S^d. Then we can numerically approximate the optimal cost-to-go functions J_k^*, $k = 0, 1, ..., N$, with the functions J_k^d computed on S^d, for example according to the method suggested in Abate et al. (2007). At the k-th iteration of this dynamic programming procedure, we can store the optimal control inputs

$$\mu_k^*(s^d) \in \arg\sup_{a \in C_a^d} \inf_{b \in C_b^d} H^d(s^d, a, b, J_{k+1}^d),\ s^d \in (W' \setminus R) \cap S^d,$$

where H^d is an appropriate discrete approximation of the operator H. This provides us with a discrete representation for an approximate max-min control policy $\mu^* = (\mu_0^*, \mu_1^*, ..., \mu_N^*)$. In particular, at time step k, $\mu_k^*(s^d)$ represent the optimal input selection over the grid volume corresponding to the grid node s^d. For the single-player case, it has also been shown in Abate et al. (2007) that the approximate control policy synthesized in such a manner provides a performance level that converges to the optimal, as the size of each grid volume is reduced.

3) Robustness and optimality: By statement (b) of Theorem 4.1, if the control were to choose the max-min policy μ^* and the adversary were to deviate from the worst-case strategy γ^*, then the reach-avoid probability will be at least $r_{s_0}^*(R, W')$. On the other hand, if the control were to deviate from the max-min policy and the adversary were to choose the worst-case strategy, then the reach-avoid probability will be at most $r_{s_0}^*(R, W')$. Thus, μ^* can be interpreted as a robust control policy in the sense that by choosing μ^*, the reach-avoid probability will be no less than $r_{s_0}^*(R, W')$, regardless of any variations in adversary strategy within the class Γ_b. It can be also interpreted as an optimal policy in the sense that it optimizes a worst-case performance index, namely the worst-case reach-avoid probability with respect to fixed choices of control policies within the class $\mathscr{M}_a$.

4) Probabilistic reach-avoid set: Consider the case in which the design specifications requires the controller to guarantee a reach-avoid probability of at least $(1 - \varepsilon)$, for some small $\varepsilon \in [0, 1)$. The set of initial conditions S_ε for which this specification is feasible can be derived from the max-min reach-avoid probability as:

$$S_0^\varepsilon = \{s_0 \in S : r_{s_0}^*(R, W') \geq (1 - \varepsilon)\}.$$

In other words, S_0^ε is the $(1 - \varepsilon)$-superlevel set of the function $s_0 \to r_{s_0}^*(R, W')$. This set can be then used to provide guidance on where the system state should be initialized. In particular, if the control were to select inputs according to the max-min policy, then the system state should be initialize inside S_0^ε in order to ensure that the desired specifications will be met.

4.4.3 Specialization to Stochastic Game Formulation of Safety Problem

As discussed in section 4.3, the solution to the probabilistic safety problem can be obtained from a complementary reach-avoid problem. In particular, for a given safe set $W \in \mathscr{B}(S)$, consider a reach-avoid problem with the value function

$$\bar{r}^*_{s_0}(S \setminus W, S) := \inf_{\mu \in \mathscr{M}_a} \sup_{\gamma \in \Gamma_b} r^{\mu,\gamma}_{s_0}(S \setminus W, S),\ s_0 \in S.$$

Then the max-min probability of safety is given by

$$p^*_{s_0}(W) = \sup_{\mu \in \mathscr{M}_a} \inf_{\gamma \in \Gamma_b} p^{\mu,\gamma}_{s_0}(W) = 1 - \bar{r}^*_{s_0}(S \setminus W, S),\ s_0 \in S. \tag{4.21}$$

By minor modifications of the proof for Theorem 4.1, it can be shown that $\bar{r}^*_{s_0}(S \setminus W, S)$ is computed by the dynamic programming recursion

$$\bar{r}^*_{s_0}(S \setminus W, S) = \mathscr{T}^N_W(\mathbf{1}_{S \setminus W})(s_0),\ s_0 \in S,$$

where the operator $\mathscr{T}_W$ is defined as

$$\mathscr{T}_W(J)(s) = \inf_{a \in C_a} \sup_{b \in C_b} \mathbf{1}_{S \setminus W}(s) + \mathbf{1}_W(s) H(s,a,b,J),\ s \in S. \tag{4.22}$$

Combining this with (4.21), we then arrive at the following result for the computation of the max-min safety probability.

Theorem 4.2. *Let $\mathscr{H}$ be a DTSHG satisfying Assumption 4.1. Let $W \in \mathscr{B}(S)$ be a Borel safe set. Then*

$$p^*_{s_0}(W) = 1 - \mathscr{T}^N_W(\mathbf{1}_{S \setminus W})(s_0),\ \forall s_0 \in S.$$

For completeness, we note that there exists an equivalent dynamic programming recursion to compute the max-min safety probability, analogous to the one given in Abate et al. (2008) for the single player case. Specifically, consider an operator $\tilde{\mathscr{T}}_W$ defined as

$$\tilde{\mathscr{T}}_W(J)(s) = \sup_{a \in C_a} \inf_{b \in C_b} \mathbf{1}_W(s) H(s,a,b,J),\ s \in S. \tag{4.23}$$

The relation between $\tilde{\mathscr{T}}_W$ and $\mathscr{T}_W$ is established through the following lemma.

Lemma 4.4. *For every $s \in S$ and $k = 0, 1, ..., N$,*

$$\tilde{\mathscr{T}}^k_W(\mathbf{1}_W)(s) = 1 - \mathscr{T}^k_W(\mathbf{1}_{S \setminus W})(s).$$

Proof. We prove this result by induction on k. The case of $k = 0$ is established by the fact that $\mathbf{1}_W = 1 - \mathbf{1}_{S\setminus W}$. Now suppose the identity holds for $k = h$, then we have for every $s \in S$,

$$\begin{aligned}
\tilde{\mathscr{T}}_W^{h+1}(\mathbf{1}_W)(s) &= \tilde{\mathscr{T}}_W(\tilde{\mathscr{T}}_W^h(\mathbf{1}_W))(s) = \tilde{\mathscr{T}}_W(1 - \mathscr{T}_W^h(\mathbf{1}_{S\setminus W}))(s) \\
&= \sup_{a\in C_a} \inf_{b\in C_b} \mathbf{1}_W(s) H(s,a,b,1 - \mathscr{T}_W^h(\mathbf{1}_{S\setminus W})) \\
&= \sup_{a\in C_a} \inf_{b\in C_b} \mathbf{1}_W(s)(1 - H(s,a,b,\mathscr{T}_W^h(\mathbf{1}_{S\setminus W}))) \\
&= \mathbf{1}_W(s) + \sup_{a\in C_a} \inf_{b\in C_b} -\mathbf{1}_W(s) H(s,a,b,\mathscr{T}_W^h(\mathbf{1}_{S\setminus W})).
\end{aligned}$$

It then follows that for every $s \in S$

$$\begin{aligned}
1 - \tilde{\mathscr{T}}_W^{h+1}(\mathbf{1}_W)(s) &= 1 - \mathbf{1}_W(s) - \sup_{a\in C_a} \inf_{b\in C_b} -\mathbf{1}_W(x) H(s,a,b,\mathscr{T}_W^h(\mathbf{1}_{S\setminus W})) \\
&= \mathbf{1}_{S\setminus W}(s) + \inf_{a\in C_a} \sup_{b\in C_b} \mathbf{1}_W(x) H(s,a,b,\mathscr{T}_W^h(\mathbf{1}_{S\setminus W})) \\
&= \mathscr{T}_W(\mathscr{T}_W^h(\mathbf{1}_{S\setminus W}))(s) = \mathscr{T}_W^{h+1}(\mathbf{1}_{S\setminus W})(s),
\end{aligned}$$

which completes the proof. □

Thus, an equivalent dynamic programming recursion for computing the max-min safety probability is given by

$$p^*_{s_0}(W) = \tilde{\mathscr{T}}_W^N(\mathbf{1}_W)(s_0),\ s_0 \in S. \tag{4.24}$$

Using either the operator $\mathscr{T}_W$ or the operator $\tilde{\mathscr{T}}_W$, we can also derive sufficient conditions of optimality for player I and II, analogous to those given in (4.14) and (4.15).

4.4.4 Analytic Reach-Avoid Example

We illustrate the sequence of steps associated with a probabilistic reachability calculation in the context of the jump Markov system example in section 4.2.1. In particular, consider a regulation problem in which the objective of player I is to drive the continuous state into a neighborhood of the origin, while staying within some safe operating region. In this case, the target set and safe set are chosen to be $R = \{q_1, q_2\} \times [-\frac{1}{4}, \frac{1}{4}]$ and $W' = \{q_1, q_2\} \times [-2, 2]$. In the following, we will solve for the max-min reach-avoid probability and player I policy over a single stage of the stochastic game ($N = 1$).

Given the DTSHG model, the operator $H(s,a,b,J)$ for a hybrid state $s = (q_1, x)$ can be derived as follows:

$$\begin{aligned}
H((q_1,x),a,b,J) &= \int_S J(s')\nu(ds'|(q_1,x),a,b) \\
&= p_1 \int_{-1}^{1} J(q_1, 2x + a + d + w)dw + \\
&\quad (1 - p_1) \int_{-1}^{1} J(q_2, 2x + a + d + w)dw.
\end{aligned} \tag{4.25}$$

For an initial condition $s_0 = (q_1, x_0)$, the max-min reach-avoid probability can be then computed as

$$r^*_{(q_1,x_0)}(R,W') = \mathscr{T}(\mathbf{1}_R)(q_1,x_0) \tag{4.26}$$
$$\begin{cases} 1, & |x_0| \leq \frac{1}{4}, \\ 0, & |x_0| > 2, \\ \sup_{a\in C_a} \inf_{b\in C_b} H((q_1,x_0),a,b,\mathbf{1}_R), & \frac{1}{4} < |x_0| \leq 2. \end{cases}$$

From equations (4.25) and (4.26), the analytic expression for the max-min reach-avoid probability in mode q_1 is:

$$r^*_{(q_1,x_0)}(R,W') = \begin{cases} 1, & |x_0| \leq \frac{1}{4} \\ \frac{1}{8}, & \frac{1}{4} < |x_0| \leq \frac{1}{2} \\ \frac{5}{8} - |x_0|, & \frac{1}{2} < |x_0| \leq \frac{5}{8} \\ 0, & |x_0| > \frac{5}{8}. \end{cases}$$

In the process of performing the dynamic programming step in (4.26), we also obtain a max-min player I policy μ^*_0 and a worst-case player II strategy γ^*_0 in mode q^1 satisfying the sufficient conditions for optimality in (4.14) and (4.15):

$$\mu^*_0(q_1,x_0) = \begin{cases} -\text{sgn}(x_0), & |x_0| > \frac{1}{2} \\ -2x_0, & |x_0| \leq \frac{1}{2}, \end{cases} \qquad \gamma^*_0((q_1,x_0),a) = \begin{cases} -1, & 2x_0 + a < 0 \\ 1, & 2x_0 + a \geq 0. \end{cases}$$

Using a similar procedure, we can compute the max-min reach-avoid probability for an initial condition $s_0 = (q_2, x_0)$ as

$$r^*_{x_0}(R,W') = \mathscr{T}(\mathbf{1}_R)(q_2,x_0) = \begin{cases} 1, & |x_0| \leq \frac{1}{4} \\ \frac{1}{8}, & \frac{1}{4} \leq |x_0| \leq 2 \\ 0, & |x_0| > 2. \end{cases}$$

Furthermore, a max-min player I policy and a worst-case player II strategy satisfying the sufficient conditions for optimality in mode q_2 can be derived as follows:

$$\mu^*_0(q_2,x_0) = \begin{cases} -\text{sgn}(x_0), & |x_0| > 2 \\ -\frac{1}{2}x_0, & |x_0| \leq 2, \end{cases} \qquad \gamma^*_0((q_2,x_0),a) = \begin{cases} -1, & \frac{1}{2}x_0 + a < 0 \\ 1, & \frac{1}{2}x_0 + a \geq 0. \end{cases}$$

As one consider more complicated system models, there may no longer be a closed-form expression for the operator $\mathscr{T}$. This would then require a numerical approximation of the dynamic programming procedure, as discussed previously in section 4.4.2.

4.5 Alternative Information Patterns

In the discussions so far, we have considered Stackelberg formulations of the probabilistic safety and reach-avoid problems, with an asymmetric information pattern which gives an advantage to Player II. As noted previously, this selection of information pattern is based upon a conservative assumption, commonly made within the context of robust control, that the intent of Player I might be available to Player II, and that player II might use this information to his/her advantage. While it is often the case that disturbances or adversaries found in practical applications will not be able to observe the actual inputs selected by the control, a control policy constructed under such an assumption is nonetheless robust to the worst-case behavior of the adversary. We will refer to problem formulations of such type as *Scenario I*.

The focus of this section is to explore several alternative information patterns that are less conservative in the assumption on player input selections. Reachability problems formulated under such settings may be of interest in application scenarios beyond those traditionally considered in robust control. In particular, they correspond to competitive or adversarial scenarios in which the control has equal or better access to information as compared with the adversary. The main results of this section are as follows. Under an asymmetric information pattern favoring player I, the Stackelberg solutions to the probabilistic safety and reach-avoid problems can be computed using a slight modification of the dynamic programming procedure from Scenario I. On the other hand, under a symmetric information pattern, the existence of Nash equilibria for the safety and reach-avoid problems in general requires randomization in the selection of player I and player II controls.

4.5.1 Stackelberg Formulation Favoring Player I

First, we consider an information pattern in which the control is allowed to select controls in response to the actions of the adversary at each stage of the dynamic process. Such a situation could for example arise in a patrol and surveillance application in which the actions of an intruder is captured by a surveillance system. Reachability problems formulated under this information pattern can be then interpreted as zero-sum Stackelberg games giving an advantage to player I in the optimization of the safety or reach-avoid probability. We refer to this type of problem formulation as *Scenario II*.

To give a more precise definition for the reachability problems in Scenario II, it is necessary to introduce the class of Markov strategies for Player I and the class of Markov policies for Player II.

Definition 4.5. A *Markov strategy* γ_a for Player I is a sequence $\gamma_a = (\gamma_0^a, \gamma_1^a, ..., \gamma_{N-1}^a)$ of universally measurable maps $\gamma_k^a : S \times C_b \to C_a$, $k = 0, 1, ..., N-1$. The set of such strategies is denoted by Γ_a.

Definition 4.6. A *Markov policy* μ_b for Player II is a sequence $\mu_b = (\mu_0^b, \mu_1^b, \mu_2^b, ...)$ of universally measurable maps $\mu_k^b : S \to C_b$, $k = 0, 1, ..., N-1$. The set of such policies is denoted by $\mathscr{M}_b$.

We briefly note that Markov policies are a subclass of Markov strategies, namely they consist of the set of Markov strategies which do not explicitly depend on the input of the other player. More specifically, $\mathscr{M}_a \subset \Gamma_a$ and $\mathscr{M}_b \subset \Gamma_b$.

Using a similar construction as in section 4.2, we can define for a given Markov strategy $\gamma_a \in \Gamma_a$ and a given Markov policy $\mu_b \in \mathscr{M}_b$ a stochastic kernel describing the closed-loop hybrid state evolution at time step k:

$$\tilde{\nu}^{\gamma_k^a,\mu_k^b}(\cdot|s_k) := \tilde{\nu}(\cdot|s_k,\gamma_k^a(s_k,\mu_k^b(s_k)),\mu_k^b(s_k)).$$

As before, this induces a probability measure, denoted by $\tilde{P}_{s_0}^{\gamma_a,\mu_b}$, on the sample space Ω. Note that if both players select Markov policies rather than Markov strategies, namely $\mu_a \in \mathscr{M}_a$ and $\mu_b \in \mathscr{M}_b$, then the probability measures in Scenario I and II are equivalent: $\tilde{P}_{s_0}^{\mu_a,\mu_b} \equiv P_{s_0}^{\mu_a,\mu_b}$.

Through the connection between the safety and reach-avoid problems as discussed in section 4.3, we will focus on the formulation of the probabilistic reach-avoid problem. Under Scenario II, the payoff function for the reach-avoid problem becomes

$$\tilde{r}_{s_0}^{\gamma_a,\mu_b}(R,W') = \tilde{E}_{s_0}^{\gamma_a,\mu_b}\left[\sum_{k=0}^{N}\left(\prod_{j=0}^{k-1}\mathbf{1}_{W'\setminus R}(s_j)\right)\mathbf{1}_K(s_k)\right] \tag{4.27}$$

where $\tilde{E}_{s_0}^{\gamma_a,\mu_b}$ denotes the expectation with respect to the probability measure $\tilde{P}_{s_0}^{\gamma_a,\mu_b}$ on the sample space Ω.

Under a zero-sum Stackelberg formulation of the reach-avoid problem, the worst-case reach-avoid probability in Scenario II under a player I strategy $\gamma_b \in \Gamma_a$ is defined as

$$\tilde{r}_{s_0}^{\gamma_a}(R,W') = \inf_{\mu_b\in\mathscr{M}_b} \tilde{r}_{s_0}^{\gamma_a,\mu_b}(R,W'),\ s_0 \in S. \tag{4.28}$$

The Stackelberg or max-min pay-off for player I in Scenario II is then given by

$$\tilde{r}_{s_0}^{*}(R,W') := \sup_{\gamma_a\in\Gamma_a} \tilde{r}_{s_0}^{\gamma_a}(R,W'),\ s_0 \in S. \tag{4.29}$$

The optimal strategy of player I and the optimal policy of player II are interpreted in terms of the Stackelberg solutions to (4.28) and (4.29), in an analogous fashion as described for the problem in Scenario I. The Stackelberg payoff in this case will be referred to as the max-min reach-avoid probability for Scenario II, while a Stackelberg solution (γ_a^*,μ_b^*) will be referred to as a max-min control strategy and a worst-case adversary policy, respectively.

A precise statement of the probabilistic reach-avoid problem in Scenario II is as follows.

Problem 4.2. Given a DTSHG $\mathscr{H}$, target set $R \in \mathscr{B}(S)$, and safe set $W' \in \mathscr{B}(S)$ such that $R \subseteq W'$:

(I) Compute the max-min reach-avoid probability for Scenario II: $\tilde{r}_{s_0}^{*}(R,W')$, $\forall s_0 \in S$;

(II) Find a max-min control strategy $\gamma_a^* \in \Gamma_a$, whenever it exists;

(III) Find a worst-case adversary policy $\mu_b^* \in \mathscr{M}_b$, whenever it exists.

Within a game theoretic context, this problem formulation can be interpreted in two different ways. From the point of view of a static optimization problem, namely the selection of a player I strategy $\gamma_a \in \Gamma_a$ and the selection of player II policy $\mu_b \in \mathscr{M}_b$ with respect to the payoff function $\tilde{r}_{s_0}^{\gamma_a,\mu_b}(R,W')$, then it is a static Stackelberg game with player I as the leader. On the other hand, from the point of view of a multi-stage dynamic game, the information structure in each stage of the dynamic game involves player I selecting inputs in response to the actions of player II. Thus, it can be also interpreted as a sequential or feedback Stackelberg game with player II as the leader. For further details, the interested reader is referred to the discussions in Breton et al. (1988) and Başar and Olsder (1999).

The problem formulations in Scenario I and Scenario II, as interpreted in terms of feedback Stackelberg games, differ only in the order of play in each stage of the dynamic game. In particular, player I goes first in Scenario I, but second in Scenario II. Thus, a dynamic programming solution to Problem 4.2 can be constructed in much the same way as described in section 4.4, except for an exchange in the order of optimization in each step of the dynamic programming procedure. More precisely, consider a dynamic programming operator $\tilde{\mathscr{T}}$ operating on Borel measurable functions from S to $[0,1]$:

$$\tilde{\mathscr{T}}(J)(s) = \inf_{b\in C_b} \sup_{a\in C_a} \mathbf{1}_R(s) + \mathbf{1}_{W'\setminus R}(s) H(s,a,b,J), s \in S. \tag{4.30}$$

In the following, we state a dynamic programming result for Scenario II. The proof is analogous to the one given in section 4.4.1 for Scenario I and is hence omitted.

Theorem 4.3. *Let $\mathscr{H}$ be a DTSHG satisfying Assumption 4.1. Let $R, W' \in \mathscr{B}(S)$ be Borel sets such that $R \subseteq W'$. Let the operator $\tilde{\mathscr{T}}$ be defined as in (4.30). Then the composition $\tilde{\mathscr{T}}^N = \tilde{\mathscr{T}} \circ \tilde{\mathscr{T}} \circ \cdots \circ \tilde{\mathscr{T}}$ (N times) is well-defined and*

(a) $\tilde{r}_{s_0}^*(R,W') = \tilde{\mathscr{T}}^N(\mathbf{1}_R)(s_0), \forall s_0 \in S$;

(b) There exists a player I strategy $\gamma_a^* \in \Gamma_a$ and a player II policy $\mu_b^* \in \mathscr{M}_b$ satisfying

$$\tilde{r}_{s_0}^{\gamma_a,\mu_b^*}(R,W') \le \tilde{r}_{s_0}^*(R,W') = \tilde{r}_{s_0}^{\gamma_a^*,\mu_b^*}(R,W') \le \tilde{r}_{s_0}^{\gamma_a^*,\mu_b}(R,W'), \tag{4.31}$$

$\forall s_0 \in S$, $\gamma_a \in \Gamma_a$, and $\mu_b \in \mathscr{M}_b$. In particular, γ_a^* is a max-min control strategy, and μ_b^* is a worst-case adversary policy.

(c) Let $J_N^* = \mathbf{1}_R$, $J_k^* = \mathscr{T}^{N-k}(\mathbf{1}_R)$, $k = 0,1,...,N-1$. If $\gamma_a^* \in \Gamma_a$ is a player I strategy which satisfies

$$\gamma_k^{a,*}(s,b) \in \arg\sup_{a\in C_a} H(s,a,b,J_{k+1}^*), \tag{4.32}$$

$\forall s \in W' \setminus R$, $b \in C_b$, $k = 0,1,\ldots,N-1$, then γ_a^* is a max-min control strategy. If $\mu_b^* \in \mathscr{M}_b$ is a player II policy which satisfies

$$\mu_k^{b,*}(s) \in \arg\inf_{b\in C_b} \sup_{a\in C_a} H(s,a,b,J_{k+1}^*), \tag{4.33}$$

$\forall s \in W' \setminus R$, $k = 0,1,\ldots,N-1$, then μ_b^* is a worst-case adversary policy.

The corresponding result for the safety problem can be derived in a similar manner as described in section 4.4.3 for Scenario I.

Given the differing interpretations of the problem formulation in Scenario II, there are correspondingly two different interpretations of the Stackelberg solution. In particular, from the point of view of a static optimization problem, the max-min control strategy γ_a^* is a selection of control strategy which optimizes the worst-case probability of achieving the reach-avoid specification over the strategy class Γ_a. This is analogous to the robustness and optimality properties of the max-min control policy in Scenario I, as discussed in section 4.4.2. On the other hand, from the perspective of a multi-stage dynamic game, one can infer from (4.32) that each component $\gamma_k^{a,*}$ of the max-min control strategy γ_a^* is a best response function with respect to selections of inputs by player II in each stage of the dynamic game.

It is intuitive that with an asymmetry in the information pattern favoring player II, the max-min reach-avoid probability in Scenario II should be equal to or higher than the max-min reach-avoid probability in Scenario I. This is confirmed by the form of the dynamic programming recursion in statement (a) of Theorem 4.3. In particular, due to the exchange in the order of optimization, we have

$$\mathscr{T}^N(\mathbf{1}_R) \leq \tilde{\mathscr{T}}^N(\mathbf{1}_R),$$

which implies that $r_{s_0}^*(R,W') \leq \tilde{r}_{s_0}^*(R,W')$, $\forall s_0 \in S$. Moreover, as will become apparent in the discussions of the following subsection, this inequality is in general strict.

4.5.2 Nash Formulation

In many practical application scenarios, it is often reasonable to assume a symmetric information pattern in which both Player I and Player II make decisions based only upon the state of the system at each time step. This is in fact the typical assumption in many competitive economic models. More generally, it is applicable within the context of a control problem in which the control and the adversary can be modeled as acting simultaneously, and hence unaware of each other's intent (de Alfaro et al., 2007). This could be for example an aircraft conflict resolution problem in which the aircraft involved only broadcast their position and heading information, but not their future intent. We will refer to reachability problem formulations with this type of information pattern as *Scenario III*.

While a symmetric information pattern may be attractive from a modeling standpoint, the existence of equilibrium player strategies, in the sense of a Nash or saddle point equilibrium (Nash, 1951), typically requires the consideration of randomized strategies. Computationally, such strategies are often significantly more difficult to synthesize as compared with non-randomized strategies, due to their large representation size. Moreover, the practical implementation of such strategies can be questionable in certain applications. For example, within the context of air traffic management, it is of interest to system operators to devise conflict resolution strategies which result in predictable aircraft behaviors.

As will be shown in this subsection, when we only consider non-randomized Markov policies in Scenario III, there does not exist in general a Nash equilibrium solution to the probabilistic safety

or reach-avoid problem. However, for such cases, the Stackelberg payoff as computed in Scenario I correspond to the lower value of the symmetric dynamic game, and hence a conservative estimate of the payoff for player I. On the other hand, if randomized Markov policies are considered, a Nash equilibrium is shown to exist, under the same set of assumptions on the DTSHG model as in Scenario I and II. In such cases, however, the problem becomes one of computation and implementation of randomized policies, as discussed above.

In order to define the problem formally, we will first specify the information structure in Scenario III. As consistent with the assumption of symmetric access to information, Player I is constrained to choose Markov policies μ_a within the class $\mathscr{M}_a$, while Player II is constrained to choose Markov policies μ_b within the class $\mathscr{M}_b$. By the discussions of the preceding subsection, under fixed choices of policies $\mu_a \in \mathscr{M}_a$ and $\mu_b \in \mathscr{M}_b$, the probability measures $P_{s_0}^{\mu_a,\mu_b}$ and $\tilde{P}_{s_0}^{\mu_a,\mu_b}$ are well-defined and equivalent, for every $s_0 \in S$. It then follows that under a fixed pair of policies (μ_a, μ_b), we have $r_{s_0}^{\mu_a,\mu_b}(R,W') = \tilde{r}_{s_0}^{\mu_a,\mu_b}(R,W')$, $\forall s_0 \in S$.

As consistent with the typical analysis procedure of symmetric zero-sum games (see for example Başar and Olsder, 1999), we now define the *lower value* and *upper value* of the reach-avoid problem in Scenario III.

Definition 4.7. The lower value of the probabilistic reach-avoid problem in Scenario III is defined as

$$r_{s_0}^{l}(R,W') := \sup_{\mu_a \in \mathscr{M}_a} \inf_{\mu_b \in \mathscr{M}_b} r_{s_0}^{\mu_a,\mu_b}(R,W'),\ s_0 \in S. \tag{4.34}$$

Definition 4.8. The upper value of the probabilistic reach-avoid problem in Scenario III is defined as

$$r_{s_0}^{u}(R,W') := \inf_{\mu_b \in \mathscr{M}_b} \sup_{\mu_a \in \mathscr{M}_a} r_{s_0}^{\mu_a,\mu_b}(R,W'),\ s_0 \in S. \tag{4.35}$$

The lower value corresponds to the case in which Player I declares his/her policy to Player II, while the upper value corresponds to the case in which Player II declares his/her policy to Player I. It can be checked that the following inequality always holds:

$$r_{s_0}^{l}(R,W') \leq r_{s_0}^{u}(R,W'),\ s_0 \in S,$$

which agrees with the intuition that the player who declares his/her policy first is at a disadvantage. Clearly, in a symmetric dynamic game, neither player would reveal his/her policy to the other ahead of time. Thus, equation (4.34) should be interpreted as a conservative calculation of the payoff from the point of view of Player I, while equation (4.35) should be interpreted as a conservative calculation of the cost from the point of view of Player II.

In the case that the upper and lower values are equal, then it may be possible to construct Nash equilibrium strategies for both players. Specifically, consider the case in which

$$r_{s_0}^{l}(R,W') = r_{s_0}^{u}(R,W'),\ \forall s_0 \in S. \tag{4.36}$$

Suppose for now that the outer supremum in (4.34) is achieved by some Markov policy $\mu_a^* \in \mathscr{M}_a$ and that the outer infimum in (4.35) is achieved by some Markov policy $\mu_b^* \in \mathscr{M}_b$. Then it can be

checked that $r_{s_0}^{\mu_a^*,\mu_b^*}(R,W') = r_{s_0}^{l}(R,W') = r_{s_0}^{u}(R,W'), \forall s_0 \in S$, and for any $\mu_a \in \mathscr{M}_a$, $\mu_b \in \mathscr{M}_b$ we have that

$$r_{s_0}^{\mu_a,\mu_b^*}(R,W') \leq r_{s_0}^{\mu_a^*,\mu_b^*}(R,W') \leq r_{s_0}^{\mu_a^*,\mu_b}(R,W'),\ \forall s_0 \in S. \tag{4.37}$$

Thus, μ_a^* can be interpreted as an optimal policy for Player I in the sense that if Player II chooses μ_b^*, then the payoff for Player I can be no greater than $r_{s_0}^{\mu_a^*,\mu_b^*}(R,W')$. In a similar manner, μ_b^* can be interpreted as an optimal policy for Player II. Using terminology from noncooperative game theory (Başar and Olsder, 1999), for any R, W' such that (4.36) holds, we say that the probabilistic reach-avoid problem in Scenario III has *value*, and any pair (μ_a^*, μ_b^*) which satisfies (4.37) is referred to as a *saddle point* or *Nash equilibrium* solution to the reach-avoid problem.

With these preliminaries, a precise statement of the probabilistic reach-avoid problem in Scenario III can be given as follows.

Problem 4.3. Given a DTSHG $\mathscr{H}$, target set $R \in \mathscr{B}(S)$, and safe set $W' \in \mathscr{B}(S)$ such that $R \subseteq W'$:

(I) Compute $r_{s_0}^{l}(R,W')$ and $r_{s_0}^{u}(R,W')$, $\forall s_0 \in S$;

(II) If the probabilistic reach-avoid problem has a value, find a saddle point solution (μ_a^*, μ_b^*).

As it turns out, there is a correspondence relation between the upper and lower value of Scenario III and the Stackelberg payoffs in Scenario I and II. This is established through the following proposition.

Proposition 4.4. *Let $\mathscr{H}$ be a DTSHG satisfying Assumption 4.1. Let $R, W' \in \mathscr{B}(S)$ be Borel sets such that $R \subseteq W'$. Then under the information pattern of Scenario III, the following identities hold:*

(a) $r_{s_0}^{l}(R,W') = r_{s_0}^{*}(R,W'),\ \forall s_0 \in S.$

(b) $r_{s_0}^{u}(R,W') = \tilde{r}_{s_0}^{*}(R,W'),\ \forall s_0 \in S.$

Proof. We will prove the result for the lower value (part (a) above), the proof for the upper value is analogous. First, for each $\mu_a \in \mathscr{M}_a$, we have by the definition of the worst-case reach-avoid probability in Scenario I that

$$r_{s_0}^{\mu_a}(R,W') = \inf_{\gamma_b \in \Gamma_b} r_{s_0}^{\mu_a,\gamma_b}(R,W') \leq \inf_{\mu_b \in \mathscr{M}_b} r_{s_0}^{\mu_a,\mu_b}(R,W'),\ \forall s_0 \in S.$$

Then it follows by the definition of the lower value in Scenario III that

$$r_{s_0}^{*}(R,W') \leq r_{s_0}^{l}(R,W'),\ \forall s_0 \in S.$$

Second, by Proposition 4.3(b), there exists a Player II strategy $\gamma_b^* \in \Gamma_b$ such that for any Player I policy $\mu_a \in \mathscr{M}_a$, we have

$$r_{s_0}^{\mu_a,\gamma_b^*}(R,W') \leq \mathscr{T}^N(\mathbf{1}_R)(s_0),\ \forall s_0 \in S.$$

For each $\mu_a \in \mathscr{M}_a$, consider a choice of Player II policy $\bar{\mu}_b \in \mathscr{M}_b$ defined by

$$\bar{\mu}_k^b(s) = \gamma_k^{b,*}(s, \mu_k^a(s)),\ s \in S,\ k = 0,1,...,N-1.$$

Then it follows that

$$\nu^{\mu_k^a, \bar{\mu}_k^b}(\cdot|s) \equiv \nu^{\mu_k^a, \gamma_k^{b,*}}(\cdot|s),\ \forall s \in S,\ k = 0,1,...,N-1,$$

From this we can deduce that for each $\mu_a \in \mathscr{M}_a$, there exists $\bar{\mu}_b \in \mathscr{M}_b$ such that, for any $s_0 \in S$, the following inequality holds

$$r_{s_0}^{\mu_a, \bar{\mu}_b}(R, W') = r_{s_0}^{\mu_a, \gamma_b^*}(R, W') \leq \mathscr{T}^N(\mathbf{1}_R)(s_0).$$

Then by the result of Theorem 4.1, we have, for each $\mu_a \in \mathscr{M}_a$ and $s_0 \in S$,

$$\inf_{\mu_b \in \mathscr{M}_b} r_{s_0}^{\mu_a, \mu_b}(R, W') \leq r_{s_0}^*(R, W'),$$

and hence $r_{s_0}^l(R, W') \leq r_{s_0}^*(R, W'),\ \forall s_0 \in S$, which completes the proof. □

In other words, the Stackelberg payoff of Scenario I can be interpreted as the lower value of Scenario III, while the Stackelberg payoff of Scenario II can be interpreted as the upper value of Scenario III. A sufficient condition for the existence of value and saddle point solution in Scenario III can be then given in terms of the dynamic programming operators in Scenario I and II.

Proposition 4.5. *Let $\mathscr{H}$ be a DTSHG satisfying Assumption 4.1. Let $R, W' \in \mathscr{B}(S)$ be Borel sets such that $R \subseteq W'$. If the operator H, as defined in (4.12), satisfies*

$$\sup_{a \in C_a} \inf_{b \in C_b} H(s,a,b,J) = \inf_{b \in C_b} \sup_{a \in C_a} H(s,a,b,J) \tag{4.38}$$

for every $s \in S$ and $J \in \mathscr{F}$, then

(a) The probabilistic reach-avoid problem in Scenario III has a value;

(b) There exists a player I policy $\mu_a^* \in \mathscr{M}_a$ and a player II policy $\mu_b^* \in \mathscr{M}_b$ such that (μ_a^*, μ_b^*) forms a saddle-point solution to the reach-avoid problem;

(c) Let $J_N^* = \mathbf{1}_R$, $J_k^* = \mathscr{T}^{N-k}(\mathbf{1}_R)$, $k = 0,1,...,N-1$. If $\mu_a^* \in \mathscr{M}_a$ is a player I policy which satisfies

$$\mu_k^{a,*}(s) \in \arg\sup_{a \in C_a} \inf_{b \in C_b} H(s,a,b,J_{k+1}^*), \tag{4.39}$$

$\forall s \in W' \setminus R,\ k = 0,1,\ldots,N-1$, and if $\mu_b^* \in \mathscr{M}_b$ is a player II policy which satisfies

$$\mu_k^{b,*}(s) \in \arg\inf_{b \in C_b} \sup_{a \in C_a} H(s,a,b,J_{k+1}^*), \tag{4.40}$$

$\forall s \in W' \setminus R,\ k = 0,1,\ldots,N-1$, then (μ_a^*, μ_b^*) is a saddle-point solution.

Proof. Suppose (4.38) holds, then we have $\mathscr{T}(J) = \tilde{\mathscr{T}}(J)$, $\forall J \in \mathscr{F}$. This in turn implies that

$$r^*_{s_0}(R,W') = \mathscr{T}^N(\mathbf{1}_R)(s_0) = \tilde{\mathscr{T}}^N(\mathbf{1}_R)(s_0) = \tilde{r}^*_{s_0}(R,W'),\ \forall s_0 \in S.$$

Combining this with Proposition 4.4, statement (a) follows.

In such a case, we have by Theorem 4.1(b) that there exists a max-min control policy $\mu^*_a \in \mathscr{M}_a$ for Scenario I satisfying

$$r^*_{s_0}(R,W') \leq r^{\mu^*_a,\gamma_b}_{s_0}(R,W'),\ \forall s_0 \in S,\ \gamma_b \in \Gamma_b.$$

Moreover, we have by Theorem 4.3(b) that there exists a worst-case adversary policy $\mu^*_b \in \mathscr{M}_b$ for Scenario II satisfying

$$\tilde{r}^{\gamma_a,\mu^*_b}_{s_0}(R,W') \leq \tilde{r}^*_{s_0}(R,W'),\ \forall s_0 \in S,\ \gamma_a \in \Gamma_a.$$

With another application of Proposition 4.4, the pair (μ^*_a,μ^*_b) can be shown to be a saddle point solution to the reach-avoid problem.

Finally, if $\mu^*_a \in \mathscr{M}_a$ is a player I policy satisfying (4.39), then μ^*_a is a max-min control policy satisfying the conditions of Theorem 4.1(b). Moreover, if $\mu^*_b \in \mathscr{M}_b$ is a player II policy satisfying (4.40), then μ^*_b is a worst-case adversary policy satisfying the conditions of Theorem 4.3(b). Statement (c) then follows. □

Equations of the form (4.38) are often referred to in literature as a minimax condition. Efforts to establish conditions for when such equations hold can be traced back to von Neumann's minimax theorem (von Neumann and Morgenstern, 1944), which has since been generalized by several authors (see for example Fan, 1953; Sion, 1958). The following set of conditions are due to Fan (1953):

Assumption 4.2.

- C_b is a compact Hausdorff space;
- For every $s \in S$, $a \in C_a$, and $J \in \mathscr{F}$, the function $b \to H(s,a,b,J)$ is lower semicontinuous and convexlike, namely for any $b_1,b_2 \in C_b$ and $\lambda \in [0,1]$, there exists $b_0 \in C_b$ such that $H(s,a,b_0,J) \leq \lambda H(s,a,b_1,J) + (1-\lambda)H(s,a,b_2,J)$.
- For every $s \in S$, $b \in C_b$, and $J \in \mathscr{F}$, the function $a \to H(s,a,b,J)$ is concavelike, namely for any $a_1,a_2 \in C_a$ and $\lambda \in [0,1]$, there exists $a_0 \in C_a$ such that $H(s,a_0,b,J) \geq \lambda H(s,a_1,b,J) + (1-\lambda)H(s,a_2,b,J)$.

Under Assumption 4.2, we have by Theorem 2 of Fan (1953) that (4.38) holds, and hence a saddle point solution exists. Intuitively speaking, this assumption requires that the function $(a,b) \to H(s,a,b,J)$ be "saddle-like" for each fixed s and J. If one were to restrict one's attention to non-randomized control policies, as in the discussions so far, this condition can be rather restrictive. In particular, it is well-known that there exists finite state games in which these conditions do not

hold and a pure strategy equilibrium does not exist. One such example, as adapted from de Alfaro et al. (2007), is given below.

Consider a two-state system with the state space $S = \{q_1, q_2\}$, and action spaces $C_a = \{1,2\}$, $C_b = \{1,2\}$. The transitions between the discrete states are described in terms of a discrete transition relation $\delta : S \times C_a \times C_b \to S$ defined as follows: $\delta(q_1, a, b) = q_1$, if $a \neq b$, and $\delta(q_1, a, b) = q_2$, otherwise; $\delta(q_2, a, b) = q_2$, $\forall a, b$. This is illustrated in Figure 4.1. The corresponding transition kernel for the DTSHG model can be derived in a straightforward manner from this diagram.

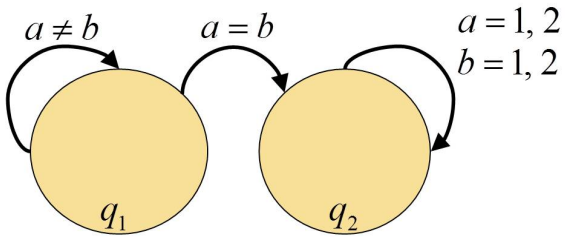


Figure 4.1: Two-state example to illustrate equilibrium strategies in symmetric dynamic games.

Now suppose that q_1 is a safe state and q_2 is an unsafe state, so that $W = \{q_1\}$. In the case that input selections are deterministic, it is intuitive that if player II were allowed to observe the inputs of player I, as in Scenario I, then player II can always select a choice of input to drive the system state into q_2 in one step. On the other hand, if player I is allowed to observe the inputs of player II, as in Scenario II, then player I can always select a choice of input at each time step to keep the system state in q_1. In particular, one can verify that if only non-randomized strategies are considered, the safety probability in Scenario I is zero, while the safety probability in Scenario II is one, over any time horizon $N \geq 1$.

On the other hand, if the players are allowed to randomize their selection of inputs, namely player I is allowed to select $a = 1$ with probability p_a and player II is allowed to select $b = 1$ with probability p_b, then one can view this as a dynamic game in which the player inputs are $p_a \in [0,1]$ and $p_b \in [0,1]$. The operator H with respect to these randomized strategies then takes on the following form:

$$H(q, p_a, p_b, J) = \begin{cases} J(q_1)(p_a + p_b - 2p_a p_b) + J(q_2)(1 - p_a - p_b + 2p_a p_b), & q = q_1 \\ J(q_2), & q = q_2. \end{cases}$$

Clearly, H is concave in p_a and convex in p_b, and hence satisfies the minimax condition (4.38). In particular, over one stage of the dynamic game, the value is 0.5 in state q_1, and the equilibrium strategies are given by $p_a = p_b = 0.5$.

In this simple example, it can be observed that randomized strategies induce a natural convexity structure in the dynamic programming operator. This is one of the primary reasons that Nash equilibria often exists only in mixed or randomized strategies, rather than in pure or non-randomized

strategies. In the following, we will proceed to show that this observation in fact holds for the probabilistic reachability problems under consideration.

Specifically, we associate with each DTSHG model $\mathscr{H} = (Q, n, C_a, C_b, \nu_x, \nu_q, \nu_r)$ a *randomized input DTSHG model* $\mathscr{H}' = (Q, n, C'_a, C'_b, \nu'_x, \nu'_q, \nu'_r)$, defined as follows:

- $C'_a = \mathscr{P}(C_a)$, $C'_b = \mathscr{P}(C_b)$;
- For every $s \in S$, $P_a \in \mathscr{P}(C_a)$, and $P_b \in \mathscr{P}(C_b)$,

$$\nu'_x(\cdot|s, P_a, P_b) := \int_{C_b} \int_{C_a} \nu'_x(\cdot|s, a, b) P_a(da) P_b(db);$$

- For every $s \in S$, $P_a \in \mathscr{P}(C_a)$, and $P_b \in \mathscr{P}(C_b)$,

$$\nu'_q(\cdot|s, P_a, P_b) := \int_{C_b} \int_{C_a} \nu'_q(\cdot|s, a, b) P_a(da) P_b(db);$$

- For every $s \in S$, $P_a \in \mathscr{P}(C_a)$, $P_b \in \mathscr{P}(C_b)$, and $q' \in Q$,

$$\nu'_r(\cdot|s, P_a, P_b, q') := \int_{C_b} \int_{C_a} \nu_r(\cdot|s, a, b, q') P_a(da) P_b(db).$$

In the above, $\mathscr{P}(C_a)$ (resp. $\mathscr{P}(C_b)$) denote the space of probability measures on C_a (resp. C_b). Since C_a and C_b are compact Borel spaces, $\mathscr{P}(C_a)$ and $\mathscr{P}(C_b)$ are also compact Borel spaces (see for example Bertsekas and Shreve, 1978, section 7.4). Furthermore, if a DTSHG model $\mathscr{H}$ satisfies Assumption 4.1, then one can verify using Proposition 7.21 of Bertsekas and Shreve (1978) that the associated randomized input model $\mathscr{H}'$ also satisfies Assumption 4.1.

The probabilistic safety and reach-avoid problems defined in terms of the model $\mathscr{H}'$ allows player policies which select inputs from randomized input spaces. In particular, player I (resp. player II) is allowed to select policies from the Markov policy space $\mathscr{M}'_a$ (resp. $\mathscr{M}'_b$) defined in terms of the input space $\mathscr{P}(C_a)$ (resp. $\mathscr{P}(C_b)$).

For a given DTSHG model $\mathscr{H}$, we say that the probabilistic reach-avoid problem in Scenario III has a value with respect to randomized policies if the reach-avoid problem defined in terms of $\mathscr{H}'$ has a value. Moreover, we call a saddle point solution (μ'_a, μ'_b) to the reach-avoid problem defined in terms of $\mathscr{H}'$ a randomized saddle point solution. In the following, we will show that under the same set of assumptions as in Scenario I and II, the reach-avoid problem in Scenario III has a randomized saddle point solution.

Proposition 4.6. *Let $\mathscr{H}$ be a DTSHG satisfying Assumption 4.1. Let $R, W' \in \mathscr{B}(S)$ be Borel sets such that $R \subseteq W'$. Then*

(a) The probabilistic reach-avoid problem in Scenario III has a value with respect to randomized policies;

(b) There exists a randomized saddle point solution to the reach-avoid problem.

Proof. Let $\mathscr{H}'$ be the randomized input DTSHG model associated with $\mathscr{H}$. Then for a given $s \in S$, $P_a \in C'_a$, $P_b \in C'_b$, and $J \in \mathscr{F}$, the operator H can be written as

$$H(s,P_a,P_b,J) = \int_{C_b}\int_{C_a}\int_S J(s')\nu(ds'|s,a,b)P_a(da)P_b(db).$$

It can be verified that H defined as above satisfies Assumption 4.2. Specifically, given that $C'_b = \mathscr{P}(C_b)$ is a compact Borel space, the first condition is satisfied. Furthermore, since $\mathscr{H}'$ satisfies Assumption 4.1, the function $(P_a,P_b) \to H(s,P_a,P_b,J)$ is continuous by the proof of Proposition 4.1. Finally, for any $P_a,\tilde{P}_a \in C'_a$ and $\lambda \in [0,1]$, we have $\hat{P}_a := \lambda P_a + (1-\lambda)\tilde{P}_a \in C'_a$ and

$$\lambda H(s,P_a,P_b,J) + (1-\lambda)H(s,\tilde{P}_a,P_b,J) = H(s,\hat{P}_a,P_b,J).$$

This implies that H is concavelike on C'_a. In a similar manner, one can show that H is convexlike on C'_b. Thus, the conditions of Assumption 4.2 are verified.

By Theorem 2 of Fan (1953), we have that the minimax condition (4.38) holds. The statements of the proposition then follows by an application of Proposition 4.5. □

4.6 Infinite Horizon Properties

In this section, we will consider infinite horizon formulations of probabilistic reachability problems. This type of formulation is particularly relevant within the context of safety problems, in which case the specifications for many practical applications are to enforce the safety property for all time (i.e. an invariance specification), rather than over some given finite time horizon. Moreover, the investigation of infinite horizon problems opens the possibility for stationary control policies. As compared with the time-varying policies generated by finite horizon reachability computations, stationary policies in general have a smaller representation size, and are easier to implement in practice.

From a theoretical perspective, there are several issues at hand when considering infinite horizon optimal control or dynamic game problems:

- The mathematical characterization of the infinite horizon payoff as the solution to an appropriate fixed-point equation;
- The convergence of the finite horizon payoffs to the infinite horizon payoff;
- The existence of optimal infinite horizon contol policies, stationary or otherwise.

In the case that the convergence property above holds, one can approximate the infinite horizon payoff through a finite horizon computation. However, as discussed in chapter 5 of Bertsekas and Shreve (1978), such a property is not always assured, and simple counterexamples can be constructed in the deterministic case. Moreover, the existence of optimal policies in infinite horizon zero-sum games can be a rather subtle and non-trivial issue when the payoff is not discounted (see

for example Kumar and Shiau, 1981; Nowak, 1985), as in the case of probabilistic safety and reach-avoid problems.

We will address these questions within the context of the probabilistic reach-avoid problem, with the understanding that results for the safety problem can be specialized from the particular case of the reach-avoid problem in which the objective is to minimize the probability of reaching a target or unsafe set. It will be shown that the infinite horizon reach-avoid probability is a fixed point of the dynamic programming operator $\mathscr{T}$ defined in section 4.4. Furthermore, this infinite horizon payoff can be approximated by the finite horizon dynamic programming procedure. In the case that the objective of the control is to maximize the reach-avoid probability, it is also shown that there exists a stationary worst-case adversary strategy. However, as consistent with results in literature (Kumar and Shiau, 1981; Nowak, 1985), the corresponding result for the control is comparatively weaker. In particular, it is shown that there exists a time-varying ε-optimal semi-Markov control policy. In the reverse case that the objective of the control is to minimize the reach-avoid probability, as in the case of the safety problem, the results are correspondingly reversed. Namely, in such a case, there exists a stationary max-min control policy.

For a precise statement of the infinite horizon reach-avoid problem, let $\mu = (\mu_0, \mu_1, \dots) \in \mathscr{M}_a$ be an infinite horizon Markov policy for player I and let $\gamma = (\gamma_0, \gamma_1, \dots) \in \Gamma_b$ be an infinite horizon Markov strategy for player II. Then by Proposition 7.28 of Bertsekas and Shreve (1978), the stochastic kernels τ^{μ_k, γ_k}, $k = 0, 1, \dots$ induce a unique probability measure $P_{s_0}^{\mu,\gamma}$ on the sample space $\Omega = \prod_{k=0}^{\infty} S$. Under a given $\mu \in \mathscr{M}_a$ and $\gamma \in \Gamma_b$, the infinite horizon reach-avoid probability is defined as

$$r_{s_0}^{\mu,\gamma}(R, W') := P_{s_0}^{\mu,\gamma}(\{(s_0, s_1, \dots) : \exists k \geq 0, (s_k \in R) \wedge (s_j \in W',\ \forall j \in [0,k])\}). \tag{4.41}$$

The above expression can be equivalently written as

$$\begin{aligned} r_{s_0}^{\mu,\gamma}(R, W') &:= P_{s_0}^{\mu,\gamma}\left(\bigcup_{k=0}^{\infty} (W' \setminus R)^k \times R\right) = \sum_{k=0}^{\infty} P_{s_0}^{\mu,\gamma}((W' \setminus R)^k \times R) \\ &= \lim_{N \to \infty} \sum_{k=0}^{N} E_{s_0}^{\mu,\gamma}\left[\left(\prod_{j=0}^{k-1} \mathbf{1}_{W' \setminus R}(x_j)\right) \mathbf{1}_R(x_k)\right] \\ &= \lim_{N \to \infty} r_{s_0}^{\mu_{0\to N}, \gamma_{0\to N}}(R, W'). \end{aligned} \tag{4.42}$$

where $\mu_{0\to N} = (\mu_0, \dots, \mu_{N-1})$ and $\gamma_{0\to N} = (\gamma_0, \dots, \gamma_{N-1})$ denote the player I policy and player II strategy, respectively, over time horizon $[0, N]$. In other words, under a fixed infinite horizon policy μ and a fixed infinite horizon strategy γ, the infinite horizon reach-avoid probability is the limit of the finite horizon reach-avoid probability as $N \to \infty$.

As in section 4.3, we define the infinite horizon worst-case reach-avoid probability under an infinite horizon player I policy $\mu \in \mathscr{M}_a$ as

$$r_{s_0}^{\mu}(R, W') = \inf_{\gamma \in \Gamma_b} r_{s_0}^{\mu,\gamma}(R, W'),\ s_0 \in S. \tag{4.43}$$

The infinite horizon max-min pay-off for player I is then given by

$$r^{\infty}_{s_0}(R,W') := \sup_{\mu \in \mathscr{M}_a} r^{\mu}_{s_0}(R,W'),\ s_0 \in S. \tag{4.44}$$

The max-min control policy and the worst-case adversary strategy are then interpreted as the Stackelberg solution to (4.43) and (4.44). Given that an optimal policy for player I may not exist in the infinite horizon case (Kumar and Shiau, 1981), we will widen the notion of optimality to ε-optimal policies. In particular, a control policy $\bar{\mu}_a \in \mathscr{M}_a$ is said to be an ε-optimal max-min control policy if

$$r^{\bar{\mu}_a}_{s_0}(R,W') \geq r^{\infty}_{s_0}(R,W') - \varepsilon,\ \forall s_0 \in S.$$

Clearly, a max-min control policy is optimal if it is 0-optimal. The definition for the worst-case adversary strategy remains the same as in the finite horizon case.

The infinite horizon reach-avoid problem for a DTSHG is stated as follows.

Problem 4.4. Given a DTSHG $\mathscr{H}$, target set $R \in \mathscr{B}(S)$, and safe set $W' \in \mathscr{B}(S)$ such that $R \subseteq W'$:

(I) Compute the infinite horizon max-min reach-avoid probability $r^{\infty}_{s_0}(R,W')$, $\forall s_0 \in S$;

(II) For a choice of $\varepsilon > 0$ such that an ε-optimal max-min control policy $\bar{\mu}_a \in \mathscr{M}_a$ exists, find such a policy.

(III) Find a worst-case adversary strategy $\gamma^* \in \Gamma_b$, whenever it exists.

In the following, it will be shown that the infinite horizon max-min reach-avoid probability is in fact a fixed-point of the dynamic programming operator $\mathscr{T}$, and that it can be approximated by the finite horizon reachability computation as described in section 4.4.1. In particular, defining the function $V^* : S \to [0,1]$ as $V^*(s_0) := r^{\infty}_{s_0}(R,W')$, $s_0 \in S$, we will show that

$$V^* = \mathscr{T}(V^*). \tag{4.45}$$

Moreover, defining the finite horizon max-min reach-avoid probability over $[0,N]$ as

$$r^{N}_{s_0}(R,W') := \sup_{\mu_{0\to N} \in \mathscr{M}_a} \inf_{\gamma_{0\to N} \in \Gamma_b} r^{\mu_{0\to N},\gamma_{0\to N}}_{s_0}(R,W'),$$

we will show that

$$r^{\infty}_{s_0}(R,W') = \lim_{N\to\infty} r^{N}_{s_0}(R,W'),\ \forall s_0 \in S. \tag{4.46}$$

By (4.42), and the definitions of $r^{\infty}_{s_0}(R,W')$ and $r^{N}_{s_0}(R,W')$, this is equivalent to showing

$$\sup_{\mu\in\mathscr{M}_a} \inf_{\gamma\in\Gamma_b} \lim_{N\to\infty} r^{\mu_{0\to N},\gamma_{0\to N}}_{s_0}(R,W') = \lim_{N\to\infty} \sup_{\mu_{0\to N}\in\mathscr{M}_a} \inf_{\gamma_{0\to N}\in\Gamma_b} r^{\mu_{0\to N},\gamma_{0\to N}}_{s_0}(R,W').$$

In other words, the limit can be exchanged with the supremum and infimum.

We begin by proving that the limit in (4.46) in fact exists.

Lemma 4.5. *For each $s_0 \in S$, the sequence $\left\{r^N_{s_0}(R,W')\right\}_{N=1}^{\infty}$ converges.*

Proof. For each $N \geq 1$, $r^N_{s_0}(R,W')$ is the finite horizon max-min reach-avoid probability over $[0,N]$. Thus, for every $s_0 \in S$ and $N \geq 1$, $r^N_{s_0}(R,W') \in [0,1]$.

By Theorem 4.1, we have that for each $N \geq 1$ and $s_0 \in S$, $r^N_{s_0}(R,W') = \mathscr{T}^N(\mathbf{1}_R)(s_0)$. From the definition of $\mathscr{T}$ in equation (4.12), it is clear that $\mathbf{1}_R \leq \mathscr{T}(\mathbf{1}_R)$. Furthermore, by the properties of integrals, it follows directly that the operator $\mathscr{T}$ satisfies a monotonicity property: if $J, J' \in \mathscr{F}$ are value functions such that $J \leq J'$, then $\mathscr{T}(J) \leq \mathscr{T}(J')$. Given these properties of $\mathscr{T}$, it can be verified that $\mathscr{T}^k(\mathbf{1}_R) \leq \mathscr{T}^{k+1}(\mathbf{1}_R)$ for every $k \geq 0$.

From this, we conclude that for each $s_0 \in S$, the sequence $\left\{r^N_{s_0}(R,W')\right\}_{N=1}^{\infty}$ is bounded and monotonically increasing, and hence converges (see for example Rudin, 1976, Theorem 3.14). □

For notational conveniences, we define a function $V_\infty : X \to [0,1]$ as

$$V_\infty(s_0) = \lim_{N\to\infty} r^N_{s_0}(R,W'),\ \forall s_0 \in S. \tag{4.47}$$

By Proposition 4.1, it follows that V_∞ is the limit of a sequence of Borel-measurable functions, and hence is also Borel-measurable (see for example Folland, 1999, Proposition 2.7). The following result shows that V_∞ is a fixed point of the operator $\mathscr{T}$.

Proposition 4.7. *Let V_∞ be defined as in (4.47) and $\mathscr{T}$ be defined as in (4.12). Then V_∞ satisfies the fixed-point equation*

$$V_\infty = \mathscr{T}(V_\infty).$$

The proof is somewhat technical and can be found in appendix B. The line of argument is adapted from that found in Kumar and Shiau (1981) for additive cost problems. In the following, we use this result to prove (4.46).

Proposition 4.8. *Let V_∞ be defined as in (4.47). Then*

$$r^\infty_{s_0}(R,W') = V_\infty(s_0),\ \forall s_0 \in S.$$

Furthermore, the function $V^* : S \to [0,1]$ defined as $V^*(s_0) := r^\infty_{s_0}(R,W')$, $s_0 \in S$ satisfies the fixed-point equation (4.45).

Proof. It can be observed from equation (4.42) that, for any fixed infinite horizon policy $\mu = (\mu_0, \mu_1, ...) \in \mathscr{M}_a$ and $\gamma = (\gamma_0, \gamma_1, ...) \in \Gamma_b$, the following inequality holds:

$$r^{\mu,\gamma}_{s_0}(R,W') \geq r^{\mu_{0\to N},\gamma_{0\to N}}_{s_0},\ \forall s_0 \in S,\ N \in \mathbb{N}.$$

This implies that, for every $s_0 \in S$ and $N \geq 1$, we have

$$\begin{aligned} r^\infty_{s_0}(R,W') &= \sup_{\mu\in\mathscr{M}_a} \inf_{\gamma\in\Gamma_b} r^{\mu,\gamma}_{s_0}(R,W') \\ &\geq \sup_{\mu_{0\to N}\in\mathscr{M}_a} \inf_{\gamma_{0\to N}\in\Gamma_b} r^{\mu_{0\to N},\gamma_{0\to N}}_{s_0}(R,W') \\ &= r^N_{s_0}(R,W'). \end{aligned}$$

It then follows that

$$r^{\infty}_{s_0}(R,W') \geq \lim_{N\to\infty} r^{N}_{s_0}(R,W') = V_{\infty}(s_0),\ \forall s_0 \in S.$$

In order to prove the reverse inequality, we define the functions $J^{\mu,\gamma}_{k\to N} : S \to [0,1]$ by

$$J^{\mu,\gamma}_{k\to N}(s_k) := r^{\mu_{k\to N},\gamma_{k\to N}}_{s_k}(R,W'),\ s_k \in S,$$

for $\mu \in \mathscr{M}_a$, $\gamma \in \Gamma_b$, and $k = 0,1,...,N-1$.

For fixed choices of Borel-measurable functions $f : S \to C_a$ and $g : S \times C_a \to C_b$, let the operator $\mathscr{T}_{f,g}$ be defined as in (4.17). Then by Lemma 4.1, for any finite horizon $[0,N]$, $\mu \in \mathscr{M}_a$, $\gamma \in \Gamma_b$, the functions $J^{\mu,\gamma}_{k\to N}$ can be computed through the backwards recursion

$$J^{\mu,\gamma}_{k\to N}(s_k) = \mathscr{T}_{\mu_k,\gamma_k}(J^{\mu,\gamma}_{k+1\to N})(s_k),\ k = 0,1,...,N-1,$$

initialized with $J^{\mu,\gamma}_{N\to N} = \mathbf{1}_R$.

Furthermore, by Proposition 4.1, there exists a Borel-measurable function $g^* : S \times C_a \to C_b$ which satisfies for every $s \in S$ and $a \in C_a$ the following identity:

$$g^*(s,a) \in \arg\inf_{b\in C_b} H(s,a,b,V_{\infty});$$

Thus, for any Borel-measurable function $f : S \to C_a$ and $s \in S$, we have

$$\begin{aligned}\mathscr{T}_{f,g^*}(V_{\infty})(s) &= \mathbf{1}_R(s) + \mathbf{1}_{W'\setminus R}(s)H(s,f(s),g^*(s,f(s)),V_{\infty})\\ &= \inf_{b\in C_b} \mathbf{1}_R(s) + \mathbf{1}_{W'\setminus R}(s)H(s,f(s),b,V_{\infty})\\ &\leq \mathscr{T}(V_{\infty})(s).\end{aligned}$$

Consider a stationary Markov strategy $\gamma^* := (g^*,g^*,...)$. We prove the following claim by backwards induction on k: for any $\mu = (\mu_0,\mu_1,...) \in \mathscr{M}_a$ and $N \geq 1$,

$$J^{\mu,\gamma^*}_{k\to N} \leq V_{\infty},\ k = 0,1,...,N.$$

For $k = N$, it can be observed that $J^{\mu,\gamma^*}_{N\to N} = \mathbf{1}_R \leq V_{\infty}$. For the inductive step, we assume the identity holds for some $k = l \in \{1,...,N\}$. Then for any $\mu \in \mathscr{M}_a$, we have

$$J^{\mu,\gamma^*}_{l-1\to N} = \mathscr{T}_{\mu_{l-1},g^*}(J^{\mu,\gamma^*}_{l\to N}) \leq \mathscr{T}_{\mu_{l-1},g^*}(V_{\infty}) \leq \mathscr{T}(V_{\infty}).$$

By Proposition 4.7, it follows that $J^{\mu,\gamma^*}_{l-1\to N} \leq V_{\infty}$, which concludes the proof of the claim.

This result implies that for every $s_0 \in S$, $\mu \in \mathscr{M}_a$, and $N \geq 1$,

$$r^{\mu_{0\to N},\gamma^*_{0\to N}}_{s_0}(R,W') = J^{\mu,\gamma^*}_{0\to N}(s_0) \leq V_{\infty}(s_0).$$

Taking the limit as $N \to \infty$, it follows that

$$\lim_{N\to\infty} r_{s_0}^{\mu_{0\to N},\gamma^*_{0\to N}}(R,W') = r_{s_0}^{\mu,\gamma^*}(R,W') \leq V_\infty(s_0),$$

for every $s_0 \in S$ and $\mu \in \mathscr{M}_a$. Thus,

$$r_{s_0}^{\infty}(R,W') = \sup_{\mu\in\mathscr{M}_a} \inf_{\gamma\in\Gamma_b} r_{s_0}^{\mu,\gamma}(R,W') \leq V_\infty(s_0),\ \forall s_0 \in S.$$

Combining this with the previous inequality, we conclude that

$$r_{s_0}^{\infty}(R,W') = V_\infty(s_0),\ \forall s_0 \in S,$$

and hence $V^* = V_\infty$. By another application of Proposition 4.7, we have that V^* is a fixed-point of the operator $\mathscr{T}$. □

In the course of proving Proposition 4.8, we have also shown the following.

Corollary 4.1. *There exists a stationary worst-case adversary strategy. In particular, if* $\gamma^* = (g^*, g^*, ...) \in \Gamma_b$ *satisfies*

$$g^*(s,a) \in \arg\inf_{b\in C_b} H(s,a,b,V_\infty),\ \forall s \in W' \setminus R, a \in C_a,$$

then γ^* *is a worst-case adversary strategy.*

In contrast with the finite horizon case, the existence of an optimal max-min control policy is not assured, due to the positive non-discounted payoff structure. The following example, as adapted from Example 1 of Kumar and Shiau (1981) provides an illustration of this fact.

Consider a Markov decision process with the state space $S = \{q_1, q_2, q_3\}$, and action spaces $C_a = [0,1]$, $C_b = 1,2$. The states q_1 and q_2 are absorbing, namely $\tau(q_1|q_1,a,b) = \tau(q_2|q_2,a,b) = 1$, $\forall a \in C_a, b \in C_b$. In state q_3, if player I chooses $a \in [0,1]$, and player II chooses $b = 1$, then the system transitions to q_1 with probability a and q_2 with probability $1-a$; on the other hand, if player II chooses $b = 2$, then the system transitions to q_1 with probability $1-a$ and remains in q_2 with probability a. This is illustrated in Figure 4.2.

Suppose q_1 is a target state and q_2 is an unsafe state, so that $R = \{q_1\}$, $W' = \{q_1, q_3\}$. By Proposition 4.8, the infinite horizon max-min reach-avoid probability is the fixed point of the equation $V^* = \mathscr{T}(V^*)$:

$$V^*(q) = \mathbf{1}_{\{q_1\}}(q) + \max_{a\in[0,1]} \min_{b\in\{1,2\}} \mathbf{1}_{\{q_3\}}(q) \sum_{q'\in X} V^*(q')\tau(q'|q,a,b).$$

Clearly, $V^*(q_1) = 1$ and $V^*(q_2) = 0$. In the case of q_3, the above can be rewritten as

$$V^*(q_3) = \max_{a\in[0,1]} \min_{b\in\{1,2\}} \tau(q_1|q_3,a,b) + V^*(q_3)\tau(q_3|q_3,a,b).$$

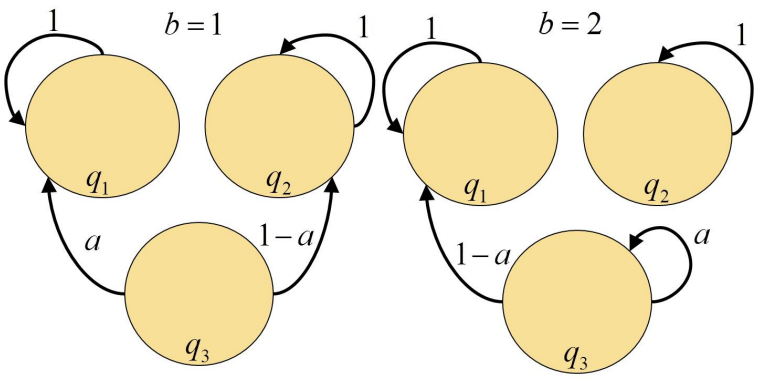


Figure 4.2: Markov chain example to illustrate infinite horizon policies.

It can be verified that the righthand side of the above equation is given by $\frac{1}{2-V^*(q_3)}$, which results in the fixed point $V^*(q_3) = 1$. However, there does not exist a Markov policy for player I which ensures that the objective of reaching q_1 from q_3 while avoiding q_2 can be achieved with probability one. Specifically, applying Corollary 4.1, we have that a worst-case adversary strategy at state q_3 is given by $\gamma^*(q_3,a) = 2$, if $a = 1$, and $\gamma^*(q_3,a) = 1$, otherwise. Under this choice of player II strategy, suppose that player I were to choose $\mu_k(q_3) = 1, \forall k \geq 0$, then $r_{q_3}^{\mu,\gamma^*}(R,W') = 0$. On the other hand, suppose that player I were to choose $\mu_l(q_3) = 1-\varepsilon$, for some $l \geq 0$ and $\varepsilon > 0$, then at time l, the system state can transition from q_3 to q_2 with probability ε, and hence $r_{q_3}^{\mu,\gamma^*}(K,K') \leq 1-\varepsilon$. This shows that there does not exist an optimal max-min control policy. In particular, observe that the policy $\mu^*(q_3) = 1$ satisfies

$$\mu^*(q_3) \in \arg\max_{a\in[0,1]} \min_{b\in\{1,2\}} \sum_{q'\in S} V^*(q')\tau(q'|q_3,a,b).$$

However, as shown above, choosing this as a stationary policy results in a reach-avoid probability strictly less than $V^*(q_3)$ (in fact, zero). There does exist, however, a stationary ε-optimal policy, namely $\mu_k(q_3) = 1-\varepsilon, \forall k \geq 0$.

The above example motivates the search for conditions under which there exists ε-optimal policies for player I. For this, we enlarge the set of control policies to those of the form $\mu = (\mu_0(s_0), \mu_1(s_0,s_1), \mu_2(s_0,s_2), \ldots)$, where μ_k depends upon both the current state s_k and also the initial condition s_0. These policies are sometimes referred to in literature as semi-Markov policies. The following result is an extension of Proposition 9.20 in Bertsekas and Shreve (1978) from the single-player case to the zero-sum game case.

Proposition 4.9. *For every $\varepsilon > 0$, there exists an ε-optimal semi-Markov max-min control policy.*

Proof. By Proposition 4.8, we have $r_{s_0}^{\infty}(R,W') = \lim_{N\to\infty} r_{s_0}^{N}(R,W')$, $\forall s_0 \in S$. For a given $\varepsilon > 0$, define the Borel sets $S_N^{\varepsilon} \subseteq S$ by

$$S_N^{\varepsilon} = \left\{ s \in S : r_{s_0}^{N}(R,W') \geq r_{s_0}^{\infty}(R,W') - \varepsilon \right\}.$$

From Lemma 4.5, it can be inferred that $S_1^\varepsilon \subseteq S_2^\varepsilon \subseteq \cdots$, and that $\cup_{N\geq 1} S_N^\varepsilon = S$. By Theorem 4.1, there exists a Markov policy $\mu^N \in \mathscr{M}_a$ for player I which satisfies

$$r_{s_0}^N(R,W') \leq r_{s_0}^{\mu^N,\gamma}(R,W'),\ \forall s_0 \in S, \gamma \in \Gamma_b.$$

Then for any initial condition $s_0 \in S_N^\varepsilon$, we have that

$$r_{s_0}^{\mu^N,\gamma}(R,W') \geq r_{s_0}^\infty(R,W') - \varepsilon,\ \forall \gamma \in \Gamma_b.$$

Now consider a policy $\hat{\mu}^N = (\mu^N, \bar{\mu}, \bar{\mu}, ...)$, where $\bar{\mu} : S \to C_a$ is arbitrary, then $\hat{\mu}^N$ is an ε-optimal policy on S_N^ε. Defining a semi-Markov policy $\tilde{\mu}$ by $\tilde{\mu} = \hat{\mu}^1$ on S_1^ε and $\tilde{\mu} = \hat{\mu}^j$ on $S_j^\varepsilon \setminus S_{j-1}^\varepsilon$, $j \geq 2$. Then $\tilde{\mu}$ is the required policy. □

In practice, one can implement the semi-Markov policy as follows. Suppose that one would like to ensure a reach-avoid probability of at least $1-\varepsilon$ over some set of initial conditions $S_0 \subset S$. Then one can perform a finite horizon reach-avoid calculation until a time instant N such that $S_0 \subseteq S_N^\varepsilon$, and apply the finite horizon optimal control policy μ^N on S_0. In the previous example, let $V_N(s_0) := r_{s_0}^N(R,W')$, $\forall s_0 \in S$, then it can be verified that $V_1(q_3) = \frac{1}{2}$ and $V_{k+1}(q_3) = \frac{1}{2-V_k(q_3)}$ for $k \geq 1$. Furthermore, the finite horizon optimal policy for player I takes the form $\mu^N = (\mu_0^N, \mu_1^N, ..., \mu_{N-1}^N)$, where $\mu_k^N(q_3) = V_{N-k}(q_3)$. For a given $\varepsilon > 0$, one could then choose an integer N sufficiently large such that $V_N(q_3) \geq 1-\varepsilon$ and implement the policy μ^N on q_3.

By the relation between the probabilistic safety and reach-avoid problems as observed in section 4.3, the infinite horizon dynamic programming results for the safety problem can be derived in an analogous fashion as in Proposition 4.7 and 4.8, except replacing the sup-inf dynamic programming operator $\mathscr{T}$ by an inf-sup dynamic programming operator. More specifically, let $W \in \mathscr{B}(S)$ be a safe set, then the infinite horizon safety probability under fixed choices of $\mu \in \mathscr{M}_a$ and $\gamma \in \Gamma_b$ is given by

$$\begin{aligned} p_{s_0}^{\mu,\gamma}(W) &:= P_{s_0}^{\mu,\gamma}(\{(s_0,s_1,...) : s_k \in W,\ \forall k \geq 0\}) \\ &= 1 - P_{s_0}^{\mu,\gamma}(\{(s_0,s_1,...) : \exists k \geq 0, (s_k \in S \setminus W)\}) \\ &= 1 - r_{s_0}^{\mu,\gamma}(S \setminus W, S), \end{aligned} \tag{4.48}$$

where $r_{s_0}^{\mu,\gamma}$ is as defined in (4.41). The infinite horizon max-min safety probability is then given by

$$p_{s_0}^*(W) := \sup_{\mu \in \mathscr{M}_a} \inf_{\gamma \in \Gamma_b} p_{s_0}^{\mu,\gamma}(W) = 1 - \bar{r}_{s_0}^*(S \setminus W, S),\ s_0 \in S. \tag{4.49}$$

where

$$\bar{r}_{s_0}^*(R,W') := \inf_{\mu \in \mathscr{M}_a} \sup_{\gamma \in \Gamma_b} r_{s_0}^{\mu,\gamma}(R,W'),\ s_0 \in S,\ R,W' \in \mathscr{B}(S). \tag{4.50}$$

The safety problem then becomes one of computing the infinite horizon minimal reach-avoid probability $\bar{r}_{s_0}^*(R,W')$ with $R = S \setminus W$ and $W' = S$, and finding an optimal control policy $\mu_a^* \in \mathscr{M}_a$ as interpreted in terms of a Stackelberg solution to (4.49).

Now consider the finite horizon minimal reach-avoid probability defined as

$$\bar{r}^N_{s_0}(R,W') := \inf_{\mu_{0\to N}\in\mathscr{M}_a} \sup_{\gamma_{0\to N}\in\Gamma_b} r^{\mu_{0\to N},\gamma_{0\to N}}_{s_0}(R,W'),$$

and an inf-sup dynamic programming operator defined as

$$\bar{\mathscr{T}}(J)(s) = \inf_{a\in C_a} \sup_{b\in C_b} \mathbf{1}_R(s) + \mathbf{1}_{W'\setminus R}(s) H(s,a,b,J),\ s\in S,\ J\in\mathscr{F}.$$

The following results can be then shown using an analogous procedure as given in the proofs of Proposition 4.7 and 4.8.

Proposition 4.10. *Let $\bar{V}_\infty : S\to[0,1]$ be defined as $\bar{V}_\infty(s_0) = \lim_{N\to\infty}\bar{r}^N_{s_0}(R,W')$ and $\bar{V}^* : S\to[0,1]$ be defined as $\bar{V}^*(s_0) := \bar{r}^*_{s_0}(R,W')$. Then*

(a) $\bar{V}^* = \bar{\mathscr{T}}(\bar{V}^*)$;

(b) $\bar{V}^* = \bar{V}_\infty$;

(c) There exists a stationary optimal control policy. In particular, if $\mu^* = (f^*, f^*, ...) \in \mathscr{M}_a$ satisfies

$$f^*(s) \in \arg\inf_{a\in C_a} \sup_{b\in C_b} H(s,a,b,\bar{V}_\infty),\ \forall s\in W'\setminus R,$$

then μ^* is an optimal control policy.

The infinite horizon safety probability can be then derived from this proposition by setting $R = S\setminus W$ and $W' = S$. It should be noted, however, that if the noise distribution in the DTSHG model has infinite support, this probability will be in general zero everywhere. Namely, given enough time, the system trajectory will eventually become unsafe. On the other hand, if one were to consider noise distributions with bounded support or alternative interpretations of the safety problem as the probability of reaching a safe set before reaching the unsafe set, for example by choosing W' in (4.50) as a strict subset of S (Hu et al., 2005), then it may be the case that the infinite horizon safety probability would no longer be identically zero and as such would be meaningful to compute.

4.7 Computational Examples

In this section, we will illustrate probabilistic reachability computation for DTSHG models through two application examples. In particular, the examples are stochastic game formulations of the aircraft conflict resolution and quadrotor target tracking examples considered previously in chapter 3 within the context of deterministic hybrid system models. The discussion here will focus on the motivations for stochastic models, the computation of the max-min probability and control policy, as well as the interpretation of the dynamic programming solutions in terms of the application of interest.

4.7.1 Aircraft Conflict Detection and Resolution

First consider the problem of two aircraft conflict resolution. As described previously in section 3.2, the relative position and heading dynamics between the two aircraft in the conflict scenario can be abstracted in terms of a deterministic, nonlinear kinematics model, with the input of aircraft 1 as the control and the input of aircraft 2 as the disturbance. A source of uncertainty which is not captured in this model, however, is the effects of wind, which can cause significant trajectory tracking errors. Such effects are difficult to model deterministically as they tend to exhibit large fluctuations from one scenario to another. Thus, they are often characterized empirically through statistical analysis of aircraft trajectory data compiled over a large number of flights (Ballin and Erzberger, 1996). This motivates the consideration of a probabilistic model of wind to augment the aircraft kinematics model.

The field of conflict detection and resolution in air traffic management features a large number of formulations and computational methods. For a comprehensive survey, the interested reader is referred to Kuchar and Yang (2000). Our approach to this problem lies at the intersection of worst-case (Tomlin et al., 2002) and probabilistic methods (Paielli and Erzberger, 1997), namely the intent of one of the aircraft is assumed to be unknown and possibly adversarial, while the wind effects on aircraft trajectory is modelled as stochastic noise. In this context, conflict detection and resolution becomes a probabilistic safety problem in which the control task is to maximize the probability of avoiding a collision between two aircraft.

We will briefly review some previous work in probabilistic conflict detection and resolution. One of the seminal works in this area is that of Paielli and Erzberger (1997), in which a model for aircraft trajectory perturbation as Gaussian noise was proposed, based upon the statistical analysis described in Ballin and Erzberger (1996). This is accompanied with an analytic method for computing the conflict probability. This formed the basis of several probabilistic conflict detection methods which followed (Prandini et al., 2000; Hwang and Seah, 2008). As more detailed trajectory models are considered, with variations to aircraft intent (Yang and Kuchar, 1997) and spatial correlation in wind effects (Hu et al., 2005), closed-form expressions for the conflict probability is often no longer available, requiring the use of numerical estimation algorithms. In comparison with these previous methods, our approach has the flexibility of being able to treat uncertainty in intent as an adversarial input rather than as a stochastic process, thus offering an interpretation of the conflict probability we compute as the probability of collision under the worst-case behavior of one of the aircraft.

To formulate the problem more precisely, let $x = (x_r, y_r, \theta_r) \in S = \mathbb{R}^2 \times [0, 2\pi]$ denote, respectively, the x-position, y-position, and heading of aircraft 2 in the reference frame of aircraft 1. By performing an Euler discretization of the kinematics equations given in section 3.2 and augmenting the resulting dynamics with a stochastic wind model as described in Hu et al. (2005), we obtain

the following model for the relative motion between two aircraft:

$$\begin{aligned} x(k+1) =& f(x(k), \omega_1(k), \omega_2(k)) + w(k) \\ =& \begin{bmatrix} x_r(k) + \Delta t(-v_1 + v_2\cos(\theta_r(k)) + \omega_1(k) y_r(k)) \\ y_r(k) + \Delta t(v_2 \sin(\theta_r(k)) - \omega_1(k) x_r(k)) \\ \theta_r(k) + \Delta t(\omega_2(k) - \omega_1(k)) \end{bmatrix} + \begin{bmatrix} w_1(k) \\ w_2(k) \\ w_3(k) \end{bmatrix}, \end{aligned} \tag{4.51}$$

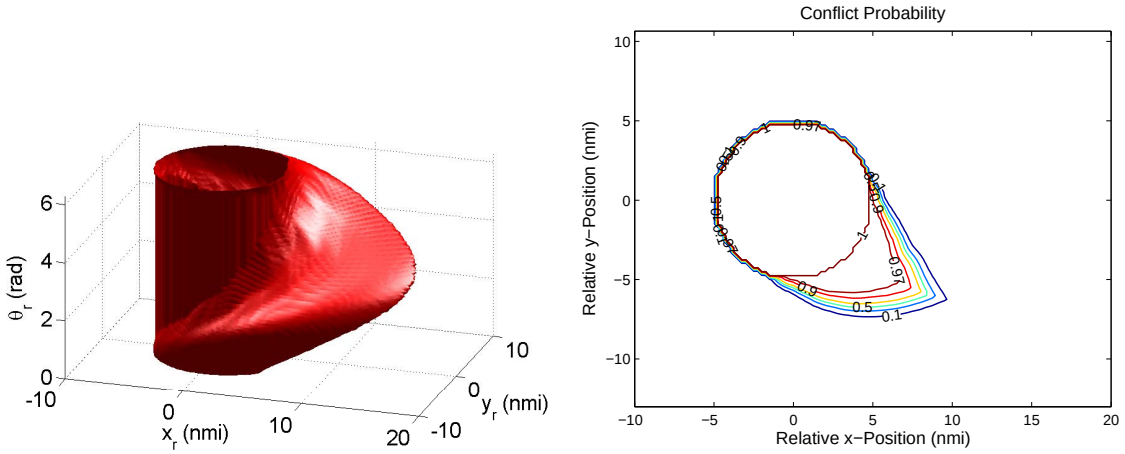
where Δt is the discretization step, v_i is the speed of aircraft i (assumed to be constant), ω_i is the angular turning rate of aircraft i, taken to be the inputs to the system. The random variables (w_1, w_2) models spatially correlated wind effects, with a Gaussian distribution $(w_1, w_2) \sim \mathcal{N}\big(0, \Sigma(x_r, y_r)\big)$, as per the wind model proposed in Hu et al. (2005). In particular, at each planar position $(x_r, y_r) \in \mathbb{R}^2$, the stochastic wind component in the stochastic differential equation (SDE) model described in Hu et al. (2005) has the distribution $\sigma_h dB(x_r, y_r, t)$ in which B is a position-dependent Brownian motion and σ_h is a positive constant. It is shown that the wind in relative coordinates has the distribution

$$\begin{aligned} w_1(t) &= \sigma_h \sqrt{2(1 - h(\|(x_r, y_r)\|))} W_1(t) \\ w_2(t) &= \sigma_h \sqrt{2(1 - h(\|(x_r, y_r)\|))} W_2(t) \end{aligned}$$

where $h : \mathbb{R} \to \mathbb{R}$ is a continuous decreasing function with $h(0) = 1$ and $\lim_{c\to\infty} h(c) = 0$ and $W(t) = (W_1(t), W_2(t))$ is a standard Brownian motion. The function h is referred to as the spatial correlation function and is chosen to be $h(c) = \exp(-\beta c)$, where β is a positive constant. The distribution of (w_1, w_2) in (4.51) is then obtained through an approximation of this distribution over one discretization step Δt. Finally, the random variable w_3 models process noise acting on the turning rate of either aircraft. It is assumed to have a Gaussian distribution $w_3 \sim \mathcal{N}(0, (\sigma_\omega \Delta t)^2)$.

As consistent with common flight maneuvers, we consider a scenario in which each aircraft is allowed to select from among one of three operation modes: straight flight, right turn, or left turn, corresponding to the angular turning rates $\omega_i = 0$, $\omega_i = -\omega$, and $\omega_i = \omega$, respectively. Here, $\omega \in \mathbb{R}$ is assumed to be a constant. The control objective of aircraft 1 is to avoid a disc D of radius R_c centered on the origin in the (x_r, y_r) plane (corresponding to a loss of minimum separation), subject to the worst-case inputs of aircraft 2. This can be then viewed as a probabilistic safety problem with the safe set given as $W = D^C \times [0, 2\pi]$. By the results of section 4.4.3, the solution to this problem can be obtained from a complementary reach-avoid problem in which the objective of aircraft 1 is to minimize the worst-case probability of entering the collision set $S \setminus W$, corresponding to the minimal reach-avoid probability $\bar{r}^*_{s_0}(S \setminus W, S)$.

For our numerical results, we choose a sampling time of $\Delta t = 15$ seconds, with a time horizon of 2.5 minutes. The radius of the protected zone is set to $R_c = 5$ nmi; the aircraft speed is set to $v_1 = v_2 = 6$ nmi per minute; and the angular turning rate is set to $\omega = 1$ degree per second. The parameters of the probability distributions are chosen as $\sigma_h = 0.5$, $\sigma_w = 0.35$, and $\beta = 0.1$. The value function is computed using a numerical discretization approach, similar to the one discussed in Abate et al. (2007), on the domain $[-10, 20] \times [-10, 10] \times [0, 2\pi]$, with a grid size of $121 \times 81 \times 73$.



(a) Set of s_0 with conflict probability $\geq 1\%$ (b) Contours of $\bar{r}^*_{s_0}(S \setminus W, S)$ at $\theta_r = \pi/2$ radians

Figure 4.3: Probability of conflict for stochastic game formulation of pairwise aircraft conflict resolution example.

The set of initial conditions s_0 for which the conflict probability is at least 1% (namely, where $\bar{r}^*_{s_0}(S \setminus W, S) \geq 0.01$) is shown in Fig. 4.3a. Outside of this set, we have a confidence level of at least 99% of avoiding a collision over a 2.5 minute time interval. A slice of the worst-case conflict probability $\bar{r}^*_{s_0}(S \setminus W, S)$ at a relative heading of $\theta_r = \pi/2$ rad is shown in Fig. 4.3b. In a conflict detection and resolution algorithm, one can use this probability map to determine the set of states at which to initiate a conflict resolution maneuver (for example where $\bar{r}^*_{s_0}$ exceeds a certain threshold), upon which time the max-min policy μ^* provides a feedback map for selecting flight maneuvers to minimize the conflict probability. A plot of this policy at a relative heading of $\theta_r = \pi/2$ rad is shown in Fig. 4.4. As can be observed, when the two aircraft are far apart, one can choose to fly straight on the intended course. However, as aircraft 2 approaches the boundary of the set shown in Fig. 4.3a, it becomes necessary for aircraft 1 to perform an evasive maneuver (turn right for the upper boundary, turn left for the lower boundary).

4.7.2 Target Tracking

Now consider the target tracking problem in which the task specification is to drive an autonomous quadrotor helicopter into a neighborhood of planar positions over a moving ground vehicle, without exceeding certain velocity limits. This problem was previously discussed in section 3.6 within a continuous time robust control framework. We describe here a stochastic formulation of the problem in which the uncertainties within the system are characterized through a mixture of deterministic bounds and stochastic noise. The motivation for this is that in an aerial robotics platform such as STARMAC, the effects of higher order dynamics and actuator noise can often be difficult to characterize through a deterministic model (Huang et al., 2009). As discussed in section 3.6, the choice of the disturbance bounds in a deterministic setting is a trade-off between robustness

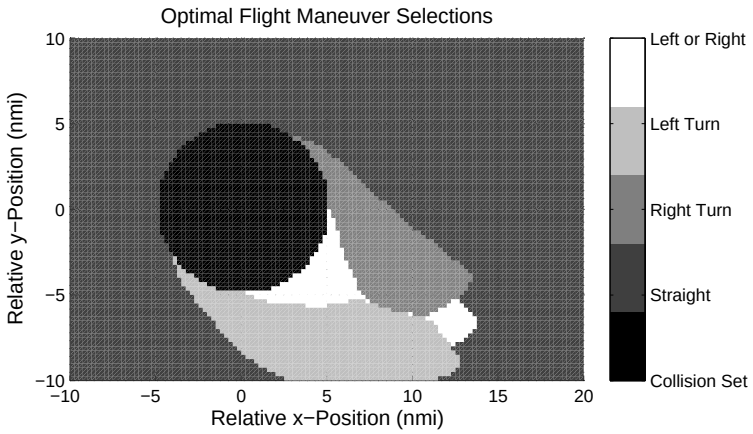


Figure 4.4: Max-min control policy at a relative heading of $\theta_r = \pi/2$ rad. The color scale is as follows: Black = collision set, dark gray = straight, medium gray = right turn, light gray = left turn, white = either left or right turn.

and feasibility. In a robust control approach, one tends to put conservative bounds on the effects of these disturbances, thus resulting in conservative control laws or sometimes even the lack of a control law which satisfies the desired motion planning specifications. To alleviate this conservatism, one may resort to disturbance bounds which captures the "majority" of the disturbance behaviors observed in practice. This, however, introduces the risk that the desired specifications may be violated, such as found in the experimental trials performed on the quadrotor platform. A probabilistic approach on the one hand provides a method for quantifying this risk, using a probabilistic model of the noise, while on the other hand allows for a relaxation of the deterministic reachability specifications.

The model of the system dynamics is obtained through a discretization of the continuous time dynamics described in section 3.6. Specifically, let x_1, x_2, y_1, y_2 denote the position and velocity of the quadrotor relative to the ground vehicle in the $x-$axis and $y-$axis, respectively. Then from the point of view of a high-level controller, the position-velocity dynamics of the quadrotor in the planar x and y directions can be modeled as decoupled double integrator, controlled in the x-direction by the roll angle ϕ and in the y-direction by the pitch angle θ angle. The corresponding equations of motion in discrete time is given by

$$\begin{aligned} x_1(k+1) =& x_1(k) + \Delta t x_2(k) + \frac{\Delta t^2}{2}(g\sin(\phi(k)) + d_x(k)) + w_1(k) \\ x_2(k+1) =& x_2(k) + \Delta t(g\sin(\phi(k)) + d_x(k)) + w_2(k) \\ y_1(k+1) =& y_1(k) + \Delta t y_2(k) + \frac{\Delta t^2}{2}(g\sin(-\theta(k)) + d_y(k)) + w_3(k) \\ y_2(k+1) =& y_2(k) + \Delta t(g\sin(-\theta(k)) + d_y(k)) + w_4(k), \end{aligned} \tag{4.52}$$

In the above, Δt is the discretization step, g is the gravitational acceleration constant, and d_x and d_y are bounded uncertainty terms corresponding to the acceleration of the ground vehicle. The

variables w_i, for $i = 1,\ldots,4$ are stochastic uncertainty terms arising from unmodeled dynamics and actuator noise. The noise variables are modeled using a Gaussian distribution, with $w_i \sim \mathcal{N}(0,(\sigma_i \Delta t)^2)$. This is based upon a simplifying modeling assumption that the noise acting on the quadrotor dynamics is generated by the sum of a large number of independent variables, in which case the Central Limit Theorem applies.

Based upon experimental trials, the bounds for the acceleration d_x and d_y of the ground vehicle are chosen to be $[-0.4, 0.4]$ m/s^2 corresponding to about 25% of the maximum allowable acceleration of the quadrotor. For this example the roll and pitch commands ϕ and θ are selected from a quantized input range due to digital implementation. Specifically, they are selected from the input range $[-10^\circ, 10^\circ]$ at a 2.5° quantization step. These quantization levels can be viewed as the discrete states of the system, similar to the discrete flight maneuvers of the previous example.

For the specification of the reach-avoid problem, the target set is chosen to be a square-shaped hover region centered on the ground vehicle, specified in (x_1, x_2) coordinates as

$$R_x = [-0.2, 0.2]m \times [-0.2, 0.2]m/s.$$

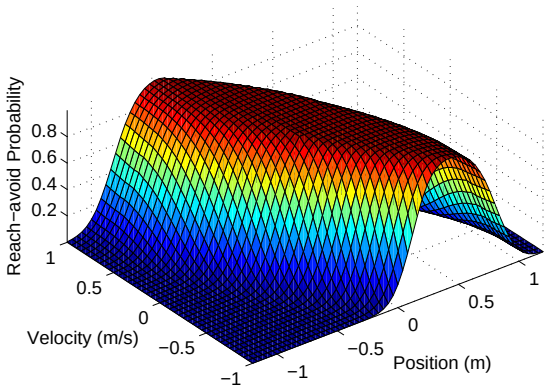
The safe set in this case is chosen to be the set of all states within the domain of interest for which the relative position remains within a desired bound and a desired velocity bound is satisfied, specified in (x_1, x_2) coordinates as

$$W'_x = [-1.2, 1.2]m \times [-1, 1]m/s.$$

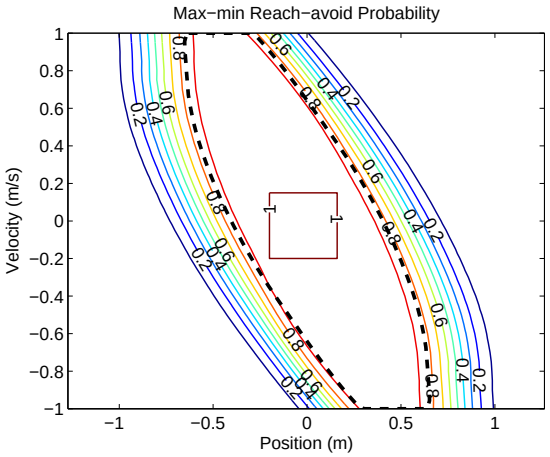
The corresponding sets in R_y and W'_y in (y_1, y_2) coordinates are chosen identically as above. The target and safe sets in two dimensions are then defined as $R = R_x \times R_y$ and $W' = W'_x \times W'_y$ respectively. Under a stochastic game formulation of the motion planning problem, the objective of the quadrotor (player I) is to reach the hover region R within finite time, while staying within the safe set W', subject to the worst-case acceleration inputs of the ground vehicle (player II).

Given that the dynamics, target set, and safe set in the x and y directions are decoupled and identical, the problem reduces to a two dimensional probabilistic reach-avoid calculation in the position-velocity space. For the numerical results to be shown here, we set the noise variance to $\sigma_i = 0.4$, the sampling time to $\Delta t = 0.1s$, and the time horizon to one second ($N = 10$). The disturbance input was discretized at $0.1m/s^2$ intervals for numerical computation. The numerical computation is performed over the safe set W'_x, on a grid size of 61×41, using a similar method as in the preceding example.

The max-min probability $r^*_{s_0}(R, W')$ of satisfying the desired motion planning objectives is shown in Fig. 4.5a over the safe set W'_x. The corresponding contours of this probability map are shown in Fig. 4.5b, with the target set R_x in the center. As a comparison, we also plot in the same figure the result of a deterministic reachability calculation from Figure 3.6 of section 3.6, characterizing the set of feasible initial conditions under the assumption that the noise obeys certain deterministic bounds. One observation is that the deterministic reach-avoid set as computed using the Hamilton-Jacobi methods described in chapter 3 bears striking resemblance to the contours of the probability map computed using the integro-difference equations described in section 4.4,



(a) Surface plot over W'_x



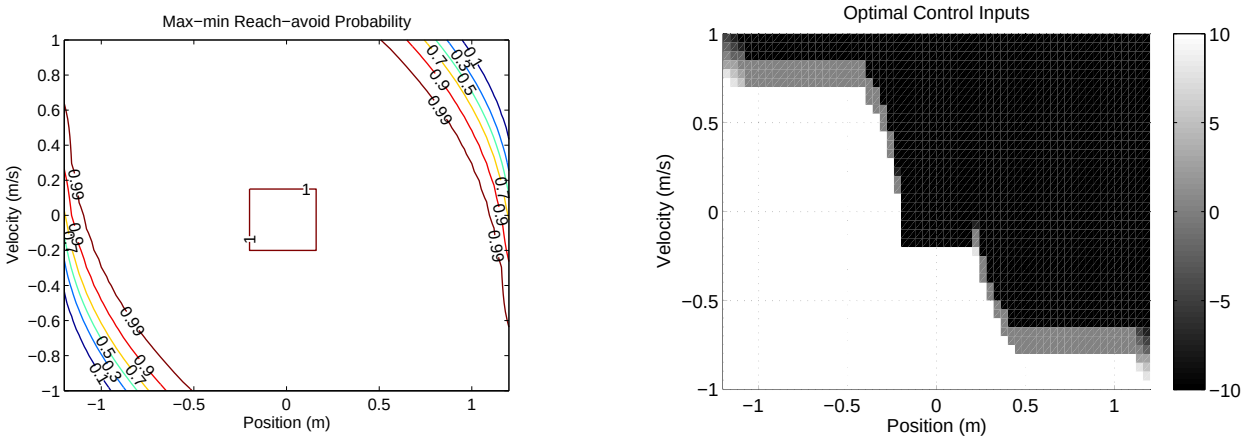
(b) Contour plot and comparison with deterministic reach-avoid set (shown in dashed outline)

Figure 4.5: Max-min reach-avoid probability $r^*_{s_0}(R,W')$ for quadrotor target tracking example with $N = 10$.

despite the relative large noise variance. In particular, the horizon-10 reach-avoid set correspond roughly to the 0.8-superlevel set of the probability map $r^*_{s_0}(R,W')$.

One interpretation of the results shown here is that the max-min probability is a quantification of the risk of violating the reach-avoid specification, if the unmodeled dynamics and disturbances behave statistically according to some noise distribution. Specifically, under the given stochastic noise model and the the probabilistic max-min control policy, system trajectories initiated from within the deterministic reach-avoid set will not always satisfy the reach-avoid specification, but rather with a probability of 80%. Another interpretation is that the probabilistic formulation is a relaxation of the deterministic reachability specification. Namely, under the unbounded noise distribution of a Gaussian model, it is impossible to synthesize, using the deterministic methods described in section 3.6, a control policy that satisfies the reach-avoid specification with probability one. On the other hand, if one were to allow the specification to be satisfied with a certain level of confidence, for example with a probability of 80%, then the max-min control policy as synthesized through the probabilistic reachability computation provides us with the feedback maps needed to enforce such a probabilistic specification.

In order to illustrate the form of the max-min control policy, as well as to investigate the infinite horizon properties of the reachability computation for this particular example, we lengthen the time horizon to $N = 40$. The resulting max-min probability $r^{40}_{s_0}(R,W')$, along with the feedback map μ^{40}_0 synthesized from this computation is shown in Figure 4.6. In this case, it was found that the reachability computation indeed exhibits a convergence behavior. In fact, over the entire safe set W', the difference between successive applications of the dynamic operator $\mathscr{T}$ at $N = 40$ was found to be no more than 4.9×10^{-5} (about 0.005%). The probability map $r^{40}_{s_0}(R,W')$ can be then interpreted as an approximation to the infinite horizon reach-avoid probability. As described in



(a) Contour plot of $r_{s_0}^{40}(R, W')$

(b) Control policy map μ_0^{40}

Figure 4.6: Max-min reach-avoid probability and control policy for quadrotor target tracking example with $N = 40$.

section 4.6, the value functions produces through this dynamic programming procedure can be also used to synthesize a max-min control policy over W' that is approximately optimal with respect to the infinite horizon payoff. Specifically, μ_0^{40} as shown in Figure 4.6 is the first feedback map in this control policy, to be applied at the first time instant $k = 0$. It is interesting to observe that this control policy has the form of a switching control policy. Namely, over large portions of W', the optimal control choice is bang-bang. On the other hand, nearing the safety constraints of W', the control law chooses an input of zero in order to prevent constraint violation. This correlates with the intuition that the control policy for a reach-avoid problem has the characteristics of a minimum time to reach control law, which was also observed experimentally in the results of section 3.6.

Chapter 5

Partial Information in Probabilistic Reachability Problems

5.1 Overview and Related Work

The controller design methods as presented in the preceding chapters are based upon an important assumption that the discrete and continuous states of the hybrid system model can be directly measured or observed. This is in fact a common assumption which appears in much of the literature on hybrid reachability problems (see for example Maler et al., 1995; Lygeros et al., 1999*b*; Asarin et al., 2000*b*; Shakernia et al., 2001; Aubin et al., 2002; Hwang et al., 2005; Koutsoukos and Riley, 2006; Gao et al., 2007; Abate et al., 2008; Tabuada, 2008; Girard et al., 2010; Mohajerin Esfahani et al., 2011), and can be reasonable as long as the state measurements or state estimates are sufficiently accurate with respect to the reachability specifications of interest. However, in the case that the measurements or estimates exhibit significant uncertainties, for example, due to limitations of what sensors can measure, imprecision in the sensor output, or measurement noise induced by the operating environment, then the reachability computation, as well as the controller synthesis procedure would need to account for the effects of decisions made under an imperfect representation of the true system state.

In this chapter, we will study probabilistic safety and reach-avoid problems within the context of a Partially Observable Discrete Time Stochastic Hybrid System (PObdtSHS), which augments the perfect information DTSHS model proposed in Amin et al. (2006) and Abate et al. (2008) with a probabilistic observation model. In particular, the possible outcomes of qualitative observations and quantitative measurements are encapsulated in an abstract observation space, while the uncertainties in the observed information are modeled in terms of a conditional probability distribution of the observations given the hybrid state. This distribution can be derived either from statistical analysis of empirical data or from statistical assumptions upon the underlying noise or disturbance, in a similar manner as the modeling of transition probabilities. Comparing with the DTSHG model of the preceding chapter, we neglect the game theoretic aspect of the reachability problem in order to focus the discussion on issues related to partial observability.

The consideration of an imperfect observation model inevitably brings up the problem of state estimation. With insight gained from classical analysis of linear and nonlinear systems (see for example Callier and Desoer, 1991; Sastry, 1999), it can be inferred that the problem of hybrid estimation is dual to the problem of hybrid control. In fact, hybrid estimation suffers from much of the same difficulties as hybrid control, for example nonlinear filtering, estimation of switching times, and estimation of continuous state under switching dynamics. In the realm of discrete state or discrete event systems, partial observability can be modeled as an output map from either the discrete state space or the discrete event space to an observation space (Ramadge, 1986; Caines et al., 1988; Ozveren and Willsky, 1990). Concepts of observability are then formulated in terms of the distinguishability of the initial condition or the current state, given the sequence of observations, much in the same way as continuous state observability. Algorithms for estimating discrete states, such as given in Caines et al. (1988) and Ozveren and Willsky (1990), typically involve maintaining a set of discrete states that are compatible with the sequence of observations. Clearly, if this set converges to a singleton, then this singleton corresponds to the exact system state.

With respect to deterministic hybrid systems, the study of observability and state estimation has largely focused on the class of linear systems with switching dynamics. When the switching behavior is assumed to be known ahead of time, then the system under analysis becomes a special class of linear time-varying systems. Observability conditions for such class of systems can be formulated through appropriate specialization of results from the study of linear time-varying systems (Ezzine and Haddad, 1988; Szigeti, 1992). In the case that the switching input is controlled, then the estimation problem can be considered dual to the switching control problem, and some conditions are given in Sun et al. (2002) for the existence of a switching input to render the system observable. The work by Vidal et al. (2003) considers a scenario in which the switching input is assumed to be an unknown piecewise constant function, and gives necessary and sufficient conditions for observability, expressed in terms of matrix rank conditions. When the switching behavior is autonomous, and the switching boundaries are assumed to be described by hyper-planes, then the system is classified as a piecewise linear or piecewise affine system. In the early work of Sontag (1981), a sufficient condition is given for the existence of a observer which uniquely determines the state of a discrete time piecewise linear system after a finite number of time steps. From a computational perspective, an algorithm is provided in Bemporad et al. (2000*a*) for checking the observability of a discrete time piecewise affine systems using the solution a mixed-integer linear program. Within a continuous time setting, the model of piecewise affine hybrid systems is considered in Collins and van Schuppen (2004), and sufficient observability conditions are given, along with procedures for constructing observers.

In the case of a stochastic hybrid system, the output trajectory corresponding to a given initial condition can vary from one execution to another, either due to process noise or measurement noise. Thus, the concept of observability does not generalize in a straightforward manner from the analysis of deterministic systems. Some efforts towards probabilistic notions of observability, however, can be found in Hwang et al. (2003) and Costa and do Val (2003), within the context of linear systems with Markov discrete transitions. The investigation into hybrid estimators for stochastic systems has its origins in deriving optimal estimators for linear Gaussian systems with a constant parameter vector taking values within a finite set (see for example Magill, 1965; Lainiotis, 1971;

Maybeck, 1982), as motivated by applications in fault detection, target tracking, and adaptive control. In generalizing this scenario to time-varying parameters, Ackerson and Fu (1970) proposed a class of linear Gaussian models in which the noise parameters are allowed to make discrete transitions according to a finite state Markov chain. This definition later evolved to encompass variations in the system matrices, and became known as Jump Linear Systems (JLS). As noted in Ackerson and Fu (1970), the optimal estimator, with respect to minimum mean square error (MMSE), is in fact a weighted sum over an exponentially growing set of Kalman filters, corresponding to the set of all possible switching sequences. Various suboptimal filtering schemes have been since proposed (a thorough review of work prior to 1982 can be found in Tugnait (1982)). Most notably, the Interacting Multiple Model (IMM) algorithm, as proposed in Blom and Bar-Shalom (1988), has found significant successes in target tracking applications (Bar-Shalom and Li., 1993). Extensions of JLS estimation algorithms to semi-Markov models with probabilistic sojourn times are discussed in Campo et al. (1991) and Petrov and Zubov (1996). Within the hybrid systems literature, an alternative approach to the hypothesis merging procedure in IMM, based upon A^* search over the set of possible discrete trajectories, is discussed in Hofbaur and Williams (2002). Furthermore, a generalization of the IMM algorithm has been proposed in Seah and Hwang (2009) for linear Gaussian models whose discrete transitions are governed by by stochastic guard conditions. Finally, as alternatives to the traditional Kalman filtering algorithms, sampling-based methods such as Markov Chain Monte Carlo algorithms (Doucet et al., 2000) and particle filtering algorithms (Koutsoukos et al., 2003; Blom and Bloem, 2007) have also been applied to various models of stochastic hybrid systems.

For control problem formulated in a partial information setting, the design of a feedback policy needs to address the following questions:

- What is the information needed for the control task at hand?
- How can we construct this information from the history of inputs and outputs?
- How do we use this information for control selection?

The first two questions relate to the estimation aspect of the problem, while the last question relate to the control aspect of the problem. In the case of a stability or regulation problem, what is needed from an estimation perspective is a convergent estimator. For deterministic systems, the existence of such an estimator is assured by sufficient conditions for observability. This is one of the reasons that much of the studies in the deterministic case has focused on finding such conditions. The practical construction of a convergent estimator, however, depends on issues of implementation. In the case of a discrete state system, the estimation algorithm as proposed in Caines et al. (1988) was used in Caines and Wang (1989) to design a dynamic observer, along with a control law for using the discrete state estimates to drive the system trajectory to a target location. In the case of linear systems, a common design for convergent observers is that of a Luenberger observer (Luenberger, 1971), which has been extended by Alessandri and Coletta (2001) and Balluchi et al. (2002) to design stabilizing controllers for hybrid systems with linear dynamics. For stochastic optimal control problems with additive cost functions, it has been shown that the estimator needed

for optimal control selection, called a sufficient statistic, is characterized by a set of Bayesian filtering equations which produces as its output the conditional distribution of the system state at each time step (see for example Bertsekas and Shreve, 1978; Kumar and Varaiya, 1986). Under mild technical assumptions, the expected value with respect to this distribution in fact coincides with the MMSE estimate, which is again a contributing reason for the study of MMSE estimators, or approximations thereof, in stochastic estimation.

Within the context of a POdtSHS model, probabilistic safety and reach-avoid problems are partial information stochastic optimal control problems with multiplicative or sum-multiplicative payoffs. Thus, they unfortunately lie beyond the common classes of control problems as described in the preceding paragraph. Perhaps the closest relative to these problems in the optimal control literature is the partial information linear exponential Gaussian (LEG) problem, whose cost is an exponential of a quadratic function, and hence multiplicative (see for example Speyer et al., 1974; Whittle, 1981; Kumar and van Schuppen, 1981; Fan et al., 1994). The close correlation of the structure of the cost with the form of the Gaussian distribution allows for the derivation of analytical solutions (which is again not the case for probabilistic reachability problems due to the indicator functions appearing in the payoff). Nonetheless, as shown in Whittle (1981), the optimal estimate for the LEG problem in fact depends on the parameters of the cost function, while the optimal control law depends on the parameters of the noise distribution. This provides an example of a partial information optimal control problem in which the type of certainty equivalence principle found in linear quadratic Gaussian (LQG) problems does not apply. Thus, as one considers non-traditional forms of cost structure, there is a need for foundational understanding of the types of estimation issues mentioned previously.

Before stating our main results, we will briefly review some previous work on partial information reachability problems. For discrete event systems, Cieslak et al. (1988) and Lin and Wonham (1988) considered problems of constructing supervisory controllers to satisfy language specifications under partial observations. The control objectives in these problems can be viewed as reachability specifications through appropriate interpretation of the language of a discrete event system under a supervisor as the closed-loop system behavior. Computational complexity issues of two-player discrete reachability games with partial information is discussed in Reif (1984). In particular, it is shown that the problem of determining the existence of a winning strategy for one of the players is in general EXPTIME-complete. An algorithm for the synthesis of winning strategies for two-player reachability games on graphs is proposed in Chatterjee et al. (2006), and is shown to achieve the EXPTIME bound in worst-case. Similar complexity results have been obtained for stochastic reachability problems in probabilistic models of two-player games with discrete state spaces and partial information (Alur et al., 1995; Bertrand et al., 2009; Gripon and Serre, 2009; Chatterjee et al., 2010). Within the class of hybrid systems referred to as linear hybrid automata, the work of Henzinger and Kopke (1999) analyzed the complexity of partial information reachability problems, and controller synthesis algorithms have been developed in Wong-Toi (1997) and De Wulf et al. (2006). Due to the simplicity of the continuous dynamics in each discrete state, as described by a differential inclusion restricted to a convex polyhedron, the techniques for analysis and control are often based upon extensions of methods developed for discrete state systems. For a discrete time hybrid system model in which the discrete state is observed, but the continuous

state measurement contains error, Del Vecchio (2009) proposes a method for designing safety controllers based upon a set-valued state estimate, computed under order preserving assumptions on the continuous dynamics. The work described in Del Vecchio et al. (2009) studies the continuous time version of this problem and provides a separation principle. The problem of discrete state estimation, under the assumption that the continuous state is measured without error, is addressed in Verma and Del Vecchio (2012). To the best of our knowledge, the results given here are some of the first of its kind on partial information reachability problems for a general class of stochastic hybrid systems.

In this chapter, we present dynamic programming solutions to the partial information probabilistic safety and reach-avoid problems for POdtSHS, as extensions of results in chapter 10 of Bertsekas and Shreve (1978) on additive cost problems. In particular, we show that by augmenting the state space with a binary random variable, the safety problem, which has a multiplicative payoff structure, is equivalent to a terminal cost problem on the augmented state space (section 5.3). The results of Bertsekas and Shreve (1978) can be then used to construct a sufficient statistic in terms of abstract filtering equations which update a conditional distribution of the augmented state. This distribution, referred to as an information state, allows us to derive an equivalent perfect state information problem and a dynamic programming algorithm for computing the optimal safety probability (section 5.4). The analysis is then extended to the reach-avoid problem, in which case it is shown that the solution is an additive cost dynamic programming algorithm on the information state space (section 5.5). We then state several consequences of these results. First, it is shown that in the case of perfect information, the class of memoryless, deterministic control policies is optimal for the safety problem within the class of randomized control policies with memory, despite the multiplicative payoff (section 5.6). This provides justification for the restriction of attention to such policies in previous work on perfect information probabilistic reachability problems (Amin et al., 2006; Abate et al., 2008; Summers and Lygeros, 2010). Second, we consider the class of Partially Observable Markov Decision Processes (POMDPs), which can be viewed as POdtSHS models with discrete state, action, and observation space (section 5.7). In this case, the filtering and policy computation can be carried out on an augmented state space with twice the number of discrete states as the original model. Third, we specialize the dynamic programming solution to hybrid system models with probability density models, in which case the sufficient statistic reduces to a set of Bayesian update equations for a conditional probability density over an augmented hybrid state space (section 5.8). The practical implementation of this solution, however, depends on the existence of a finite dimensional representation for the conditional density.

5.2 Model and Problem Formulation

5.2.1 Partially observable Discrete Time Stochastic Hybrid System

The model for a partially observable discrete time stochastic hybrid system (POdtSHS) augments the DTSHS model proposed in Amin et al. (2006) and Abate et al. (2008) with an observation space and a stochastic observation model. It can be viewed as a particular instantiation of the imperfect

state information model given in Bertsekas and Shreve (1978).

Definition 5.1 (POdtSHS)**.** A partially observable discrete time stochastic hybrid system is a tuple $\mathscr{H} = (Q, n, C_a, Z, \nu_x, \nu_q, \nu_r, \zeta_0, \zeta)$, defined as follows.

- *Discrete state space* $Q := \{q_1, q_2, ..., q_m\}$, $m \in \mathbb{N}$;
- *Dimensions of continuous state space* $n : Q \to \mathbb{N}$: a map which assigns to each discrete state $q \in Q$ the dimension of the continuous state space $\mathbb{R}^{n(q)}$. The hybrid state space is given by $S := \bigcup_{q \in Q} \{q\} \times \mathbb{R}^{n(q)}$;
- *Control input space* C_a: a nonempty Borel space;
- *Observation space* Z: a nonempty Borel space;
- *Continuous state transition kernel* $\nu_x : \mathscr{B}(\mathbb{R}^{n(\cdot)}) \times S \times C_a \to [0,1]$: a Borel-measurable stochastic kernel on $\mathbb{R}^{n(\cdot)}$ given $S \times C_a$, which assigns to each $s = (q,x) \in S$ and $a \in C_a$ a probability measure $\nu_x(\cdot|x,a)$ on the Borel space $(\mathbb{R}^{n(q)}, \mathscr{B}(\mathbb{R}^{n(q)}))$;
- *Discrete state transition kernel* $\nu_q : Q \times S \times C_a \to [0,1]$: a Borel-measurable discrete stochastic kernel on Q given $S \times C_a$, which assigns to each $s \in S$ and $a \in C_a$ a probability distribution $\nu_q(\cdot|s,a)$ over Q;
- *Reset transition kernel* $\nu_r : \mathscr{B}(\mathbb{R}^{n(\cdot)}) \times S \times C_a \times Q \to [0,1]$: a Borel-measurable stochastic kernel on $\mathbb{R}^{n(\cdot)}$ given $S \times C_a \times Q$, which assigns to each $s \in S$, $a \in C_a$, and $q' \in Q$ a probability measure $\nu_r(\cdot|s,a,q')$ on the Borel space $(\mathbb{R}^{n(q')}, \mathscr{B}(\mathbb{R}^{n(q')}))$.
- *Initial observation kernel* $\zeta_0 : \mathscr{B}(Z) \times S \to [0,1]$: a Borel-measurable stochastic kernel on Z given S, which assigns to each $s \in S$ a probability measure $\zeta_0(\cdot|s)$ on the Borel space $(Z, \mathscr{B}(Z))$;
- *Observation kernel* $\zeta : \mathscr{B}(Z) \times S \times C_a \to [0,1]$: a Borel-measurable stochastic kernel on Z given $S \times C_a$, , which assigns to each $s \in S$ and $a \in C_a$ a probability measure $\zeta(\cdot|s,a)$ on the Borel space $(Z, \mathscr{B}(Z))$;

The definitions for an abstract observation space Z and observation kernels ζ_0 and ζ are based upon the abstract model for an imperfect state information stochastic optimal control problem presented in chapter 10 of Bertsekas and Shreve (1978). The generality of these definitions can be used to treat a wide range of observation models found in practice. In particular, a discrete observation space $O := \{o_1, o_2, ..., o_{m'}\}$ and a Euclidean observation space $\mathbb{R}^{n_o}$ are both Borel spaces. Within this context, the observation kernel ζ can be interpreted as the conditional distribution of the discrete or continuous observations given a hybrid state $s \in S$ and control input $a \in C_a$. By the analysis given in Davis (1993), we also have that the hybrid state space defined by $Z = O \times \bigcup_{q \in Q} \mathbb{R}^{n_o(q)}$, where $n_o : Q \to \mathbb{N}$ is the dimension of the continuous observation space in each discrete state, is a Borel space. In this case, the kernel ζ can be interpreted as the joint conditional distribution

of the discrete and continuous observations given the hybrid state and control input. As a consequence, it is worth noting that the perfect state information DTSHS model in Amin et al. (2006); Abate et al. (2008) can be considered a special class of the POdtSHS by specifying $Z = S$ and $\zeta_0(dz|s) = \zeta(dz|s,a) = \delta_s$, where δ_s is a probability measure on S which assigns probability mass one to the point s. To illustrate this modeling framework, consider the partially observable jump linear system shown in Figure 5.1.

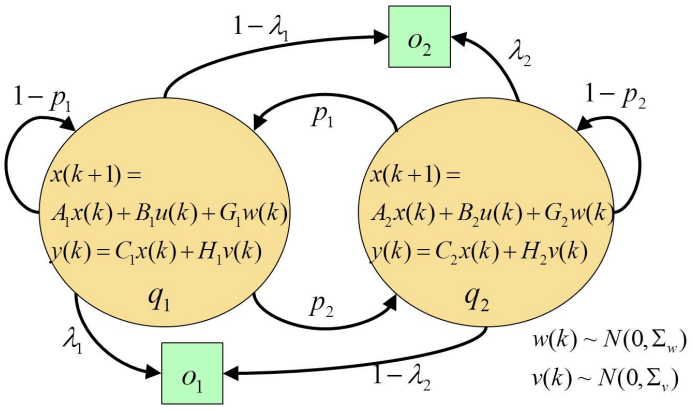


Figure 5.1: Jump linear system example to illustrate POdtSHS modeling framework.

Here we have $x \in \mathbb{R}^n$, $u \in \mathbb{R}^{n_i}$, $y \in \mathbb{R}^{n_o}$, $w \in \mathbb{R}^{n_w}$, $v \in \mathbb{R}^{n_v}$, and A_i, B_i, C_i, G_i, H_i, Σ_v, Σ_w are matrices of appropriate dimensions. In this example, the hybrid state space is given by $S = \{q_1, q_2\} \times \mathbb{R}^n$, the control input space is given by $C_a = \mathbb{R}^{n_i}$, and the observation space is given by $Z = O \times \mathbb{R}^{n_o}$, where $O = \{o_1, o_2\}$. The discrete transition kernel ν_q can be derived as $\nu_q(q_j|(q_i,x),u) = p_i$, if $j = i$, and $\nu_q(q_j|(q_i,x),u) = 1 - p_i$, otherwise. The continuous and reset transition kernels are described as $\nu_x(dx'|(q_i,x),u) = \nu_r(dx'|(q_i,x),u,q_j) \sim \mathcal{N}(A_ix + B_iu, G_i\Sigma_wG_i^T)$. Finally, the observation kernels are given by $\zeta_0(o,dy|(q,x)) = \zeta(o,dy'|(q,x),u) = \zeta_o(o|q)\zeta_x(dy|(q,x))$, where $\zeta_x(dy|(q_i,x)) \sim \mathcal{N}(C_ix, H_i\Sigma_vH_i^T)$ and $\zeta_o(o_j|q_i) = \lambda_i$, if $j = i$, and $\zeta_o(o_j|q_i) = 1 - \lambda_i$, otherwise.

Under a PDTSHS model, the available information at each time step k is the observation and input history $(z(0), a(0), ..., z(k-1), a(k-1), z(k))$, along with the probability distribution of the initial state $(q(0), x(0)$. For compactness of notation, we define as in Bertsekas and Shreve (1978) the information spaces

$$I_k = Z^{k+1} \times C_a^k,\ k = 0, 1, ...$$

An element of I_k is called the information vector at time step k. For the initial state distribution, we denote the set of probability measures on S by $\mathscr{P}(S)$. By Corollary 7.25.1 of Bertsekas and Shreve (1978), $\mathscr{P}(S)$ is also a Borel space. To keep the discussions general, we consider the set of randomized control policies depending on the initial state distribution and information vector at time k.

Definition 5.2. A policy π' for $\mathscr{H}$ is a sequence $\pi' = (\pi'_0, \pi'_1, ..., \pi'_{N-1})$ of universally measurable stochastic kernels $\pi'_k : \mathscr{B}(C_a) \times \mathscr{P}(S) \times I_k \to [0,1]$, which assigns to each initial distribution

$p_0 \in \mathscr{P}(S)$ and information vector $i_k \in I_k$ a probability measure $\pi'_k(da|p_0;i_k)$ on the Borel space $(C_a, \mathscr{B}(C_a))$. The set of such policies is denoted by Π'.

If for each k, initial distribution $p_0 \in \mathscr{P}(S)$ and information vector $i_k \in I_k$ the stochastic kernel π'_k assigns probability mass one to some point in C_a, the policy π' is said to be *non-randomized*. The class of non-randomized policies for $\mathscr{H}$ is denoted as Π. For any $\pi' \in \Pi$, we can identify the stochastic kernels π'_k, $k = 0,1,...,N-1$ with a sequence of universally measurable maps $\pi_k : \mathscr{P}(S) \times I_k \to C_a$, $k = 0,1,...,N-1$ (see for example Bertsekas and Shreve, 1978, Corollary 7.44.3). As shown in Bertsekas and Shreve (1978), for an additive cost imperfect state information stochastic optimal control problem, it is sufficient to consider the class of non-randomized policies Π over the set of general policies Π'. It turns out that a similar result also holds for the probabilistic safety problem, which has a multiplicative cost structure.

Using a similar procedure as described in section 4.2 for a DTSHG, one can construct from ν_x, ν_q, and ν_r a Borel-measurable stochastic kernel $\nu : \mathscr{B}(S) \times S \times C_a \to [0,1]$ describing the hybrid state evolution at each time step. With these definitions, the execution of the POdtSHS under a given initial distribution $p_0 \in \mathscr{P}(S)$ and policy $\pi' \in \Pi'$ is as described in Algorithm 5.2.1.

Algorithm 5.2.1 POdtSHS Execution

Require: Initial distribution $p_0 \in \mathscr{P}(S)$ and control policy $\pi' \in \Pi'$.
  Extract from S a value s_0 according to p_0;
  Extract from Z a value z_0 according to $\zeta_0(\cdot|s_0)$;
  Set $s(0) = s_0$ and $i_0 = y_0$;
  for $k = 0$ to $N-1$ **do**
    Extract from C_a a value a_k for $a(k)$ according to $\pi'_k(\cdot|p_0;i_k)$;
    Extract from S a value s_{k+1} for $s(k+1)$ according to $\nu(\cdot|s_k,a_k)$;
    Extract from Z a value z_{k+1} for $z(k+1)$ according to $\zeta(\cdot|s_{k+1},a_k)$;
    Set $i_{k+1} = (i_k, a(k), z(k+1))$;
  end for
  return Sample Path $\{(s_0, z_0, a_0, ..., s_{N-1}, z_{N-1}, a_{N-1}, s_N, z_N)\}$.

Now consider the sample space of state, observation, and control sequences over k time steps given by $\Omega_k := S^{k+1} \times Z^{k+1} \times C_a^k$, equipped with the canonical product topology $\mathscr{B}(\Omega_k) := \prod_{j=1}^{k+1}(\mathscr{B}(S) \times \mathscr{B}(Z)) \times \prod_{j=1}^{k} \mathscr{B}(C_a)$. Then by Proposition 7.45 of Bertsekas and Shreve (1978), for a given initial distribution $p_0 \in \mathscr{P}(S)$ and policy $\pi' \in \Pi'$, the stochastic kernels ν, ζ_0, and ζ induce a unique probability measure $P_k(\pi, p)$ on Ω_k. In particular, on measurable rectangles

which generates the Borel σ-algebra $\mathscr{B}(\Omega_k)$, $P_k(\pi, p_0)$ is defined as

$$
\begin{aligned}
&P_k(\pi', p_0)(\prod_{j=0}^{k-1}(S_j \times Z_j \times C_{a,j}) \times S_k \times Z_k) \qquad (5.1)\\
&= \int_{S_0}\int_{Z_0}\int_{C_{a,0}} \cdots \int_{S_{k-1}}\int_{Z_{k-1}}\int_{C_{a,k-1}}\int_{S_k}\int_{Z_k} \zeta(dz_k|s_k,a_{k-1})\nu(ds_k|s_{k-1},a_{k-1})\\
&\times \pi'_{k-1}(da_{k-1}|p;i_{k-1})\zeta(dz_{k-1}|s_{k-1},a_{k-2})\nu(ds_{k-1}|s_{k-2},a_{k-2})\\
&\times \cdots \pi'_0(da_0|p;z_0)\zeta_0(dz_0|s_0)p_0(ds_0),
\end{aligned}
$$

where $S_0,...,S_k \in \mathscr{B}(S)$, $Z_0,...,Z_k \in \mathscr{B}(Z)$, and $C_{a,0},...,C_{a,k-1} \in \mathscr{B}(C_a)$ are Borel subsets of the state space, observation space, and control input space, respectively. In the following, we describe how this probability measure allows us to quantify the probability of safety for a POdtSHS.

5.2.2 Partial Information Safety Problem

Now consider the probabilistic safety problem. Assume that a Borel set $W \in \mathscr{B}(S)$ is given as a safe set. The probability that the hybrid state trajectory $(s_0, s_1, ..., s_N)$ remains in W for an initial distribution $p_0 \in \mathscr{P}(S)$ and $\pi' \in \Pi'$ is given by

$$
\begin{aligned}
p^{\pi'}(p_0;W) &:= P_N(\pi', p_0)(\{(s_0,z_0,a_0,...,s_{N-1},z_{N-1},a_{N-1},s_N,z_N) : s_k \in W,\ \forall k \in [0,N]\})\\
&= P_N(\pi', p_0)(W^{N+1} \times Z^{N+1} \times C_a^N). \qquad (5.2)
\end{aligned}
$$

By (5.1) and Proposition 7.45 of Bertsekas and Shreve (1978), the safety probability in (5.2) can be rewritten as

$$
\begin{aligned}
p^{\pi'}(p_0;W) &= \int_W\int_Z\int_{C_a} \cdots \int_W\int_Z\int_{C_a}\int_W\int_Z \zeta(dz_N|s_N,a_{N-1})\nu(ds_N|s_{N-1},a_{N-1})\\
&\quad \times \pi'_{N-1}(da_{N-1}|p;i_{N-1})\zeta(dz_{N-1}|s_{N-1},a_{N-2})\nu(ds_{N-1}|s_{N-2},a_{N-2})\\
&\quad \times \cdots \pi'_0(da_0|p;z_0)\zeta_0(dz_0|s_0)p_0(ds_0),\\
&= \int_{\Omega_N}\prod_{k=0}^{N}\mathbf{1}_W(s_k)dP_N(\pi', p_0) = E_{p_0}^{\pi'}\left[\prod_{k=0}^{N}\mathbf{1}_W(s_k)\right], \qquad (5.3)
\end{aligned}
$$

where $E_{p_0}^{\pi'}$ denotes the expectation with respect to the probability measure $P_N(\pi', p_0)$ on the sample space Ω_N.

Our control objective is to maximize this probability over the general policy space Π'. More precisely, the problem statement is as follows:

Problem 5.1. Given a POdtSHS $\mathscr{H}$, initial distribution $p_0 \in \mathscr{P}(S)$, and safe set $W \in \mathscr{B}(S)$:

1. Compute the maximal probability of safety

$$p^*(p_0;W) := \sup_{\pi' \in \Pi'} p^{\pi'}(p_0;W);$$

2. Find an optimal policy $\pi^* \in \Pi'$, if it exists, such that $p^*(p_0;W) = p^{\pi^*}(p_0;W)$. Otherwise, for a choice of $\varepsilon > 0$, find an ε-optimal policy $\pi^*_\varepsilon \in \Pi'$ satisfying

$$p^{\pi^*_\varepsilon}(p_0;W) \geq p^*(p_0;W) - \varepsilon.$$

5.3 Sufficient Statistics and Equivalent Perfect State Information Problem

It is well-known in the stochastic optimal control literature that under an additive cost structure, the imperfect state information problem can be converted into one of perfect state information through the notion of sufficient statistics, which is, roughly speaking, an estimator which provides enough information to allow optimal control selection with respect to the history of observations and controls (see for example Bertsekas and Shreve, 1978, chapter 10). Under mild assumptions, the sufficient statistic for an additive cost problem can be shown to be the conditional probability distribution of the system state given the information vector. However, due to the multiplicative cost structure of the probabilistic safety problem, it is no longer sufficient to maintain a conditional distribution of the current state, but also some information about the history of state evolution. As will be shown in this section, a sufficient statistic for our problem consists of a filtered estimate of the current state, along with an augmented state variable which keeps track of whether the state history has remained within the safe set W.

We will proceed in several steps. First, the POdtSHS will be augmented with an auxiliary state, so as to enable an equivalent terminal cost formulation of Problem 5.1. Second, using this terminal cost formulation, a statistic sufficient for control will be constructed from the results given in Bertsekas and Shreve (1978). Finally, an equivalent perfect state information problem will be formulated through the sufficient statistic.

5.3.1 Terminal Cost Problem on Augmented System

As a first step, we augment the POdtSHS model of the previous section with the binary random variables $h_k : \Omega_k \to \{0,1\}$, $k = 0,1,...,N$, defined as:

$$h_0 := 1;\ h_k := \prod_{j=0}^{k-1} \mathbf{1}_W(s_j),\ k \geq 1. \tag{5.4}$$

For the rest of this chapter, we will refer to h_k as the *history state*. Now consider an augmented POdtSHS model with the expanded state space $\tilde{S} = \{0,1\} \times S$, in which the state of the system at any time k is given by the pair (h_k, s_k). From (5.4), the history state can be recursively updated as

$$h_{k+1} = \mathbf{1}_W(s_k)h_k,\ h_0 = 1,$$

which results in an augmented state transition kernel $\tilde{\nu} : \mathscr{B}(\tilde{S}) \times \tilde{S} \times C_a \to [0,1]$ defined as follows:

$$\tilde{\nu}((h_{k+1}, ds_{k+1})|(h_k, s_k), a_k) = \begin{cases} \nu(ds_{k+1}|s_k, a_k), & h_k = 0, h_{k+1} = 0 \\ 0, & h_k = 0, h_{k+1} = 1 \\ \mathbf{1}_{S\setminus W}(s_k)\nu(ds_{k+1}|s_k, a_k), & h_k = 1, h_{k+1} = 0 \\ \mathbf{1}_W(s_k)\nu(ds_{k+1}|s_k, a_k), & h_k = 1, h_{k+1} = 1. \end{cases} \tag{5.5}$$

Similarly, we can define the observation kernels $\tilde{\zeta}_0 : \mathscr{B}(Z) \times \tilde{S} \to [0,1]$ and $\tilde{Z} : \mathscr{B}(Z) \times \tilde{S} \times C_a \to [0,1]$ on the extended state space:

$$\tilde{\zeta}_0(dz_k|h_k, s_k) = \zeta_0(dz_k|s_k), \tag{5.6}$$

$$\tilde{\zeta}(dz_k|h_k, s_k, a_{k-1}) = \zeta(dz_k|s_k, a_{k-1}). \tag{5.7}$$

Clearly, $\tilde{\nu}$, $\tilde{\zeta}_0$, and $\tilde{\zeta}$ are Borel-measurable. We denote the augmented POdtSHS model by $\tilde{\mathscr{H}} := (S, C_a, Z, \tilde{\nu}, \tilde{\zeta}_0, \tilde{\zeta})$.

Now consider a Borel-measurable function $\xi : \mathscr{P}(S) \to \mathscr{P}(\tilde{S})$ which takes an initial state distribution on S to an initial state distribution on $\tilde{S}$:

$$\xi(p_0)(h_0, ds_0) = \begin{cases} 0, & h_0 = 0 \\ p_0(ds_0), & h_0 = 1. \end{cases} \tag{5.8}$$

Clearly, ξ is one-to-one. As such, by Kuratowski's theorem (see for example Bertsekas and Shreve, 1978, Proposition 7.15), $\mathscr{P}(S)$ and $\xi(\mathscr{P}(S)) \subset \mathscr{P}(\tilde{S})$ are isomorphic Borel spaces, with the Borel isomorphism ξ.

We define the set of admissible control policies for an augmented POdtSHS model $\tilde{\mathscr{H}}$ as follows.

Definition 5.3. A policy $\tilde{\pi}'$ for $\tilde{\mathscr{H}}$ is a sequence $\tilde{\pi}' = (\tilde{\pi}'_0, \tilde{\pi}'_1, ..., \tilde{\pi}'_{N-1})$ of universally measurable stochastic kernels $\tilde{\pi}'_k : \mathscr{B}(C_a) \times \xi(\mathscr{P}(S)) \times I_k \to [0,1]$, which assigns to each initial distribution $\xi(p_0)$ and information vector $i_k = (z_0, a_0, ..., z_{k-1}, a_{k-1}, z_k)$ a probability measure $\tilde{\pi}'_k(da_k|\xi(p_0); i_k)$ on the Borel space $(C_a, \mathscr{B}(C_a))$. The set of such policies is denoted by $\tilde{\Pi}'$.

Similarly as before, we denote the class of non-randomized policies for $\tilde{\mathscr{H}}$ as $\tilde{\Pi}$. Given that ξ is a Borel isomorphism, $\tilde{\Pi}'$ and Π' can be viewed as identical policy spaces. In particular, a policy $\tilde{\pi}' = (\tilde{\pi}'_0, \tilde{\pi}'_1, ..., \tilde{\pi}'_{N-1})$ can be identified with a policy $\pi' = (\pi'_0, \pi'_1, ..., \pi'_{N-1}) \in \Pi'$ through the relation

$$\tilde{\pi}'_k(da_k|\xi(p_0); i_k) = \pi'_k(da_k|p_0; i_k). \tag{5.9}$$

For a given initial distribution $\xi(p_0) \in \xi(\mathscr{P}(S))$ and policy $\tilde{\pi}' \in \tilde{\Pi}'$, the stochastic kernels $\tilde{\nu}$, $\tilde{\zeta}_0$, and $\tilde{\zeta}$ induce a unique probability measure $\tilde{P}_k(\tilde{\pi}', \xi(p_0))$ on the sample space $\tilde{\Omega}_k := \tilde{S}^{k+1} \times Z^{k+1} \times C_a^k$ defined similarly as in (4.1). Using this probability measure, we consider a probabilistic reachability problem on the augmented system. Specifically, let $W \in \mathscr{B}(S)$ be a safe set. Then

given an initial distribution $\xi(p_0) \in \xi(\mathscr{P}(S))$ and policy $\tilde{\pi}' \in \tilde{\Pi}'$, the probability that the trajectory $(\tilde{s}_0, \tilde{s}_1, ..., \tilde{s}_N)$ terminates in a state such that $h_N = 1$ and $s_N \in W$ is given by

$$\begin{aligned}\tilde{p}^{\tilde{\pi}'}(\xi(p_0); \{1\} \times W) &:= \tilde{P}_N(\tilde{\pi}', \xi(p_0))(\{(h_0, s_0, z_0, a_0, ..., h_N, s_N, z_N) : h_N = 1 \wedge s_N \in W\}) \\ &= \tilde{P}_N(\tilde{\pi}', \xi(p_0))(\{0,1\}^N \times \{1\} \times S^N \times W \times Z^{N+1} \times C_a^N) \\ &= E^{\tilde{\pi}'}_{\xi(p_0)}\left[\mathbf{1}_{\{1\} \times W}(\tilde{s}_N)\right]. \end{aligned} \tag{5.10}$$

Now consider a probabilistic reachability problem for $\tilde{\mathscr{H}}$ defined as follows.

Problem 5.2. Given an initial distribution $p_0 \in \mathscr{P}(S)$ and an augmented POdtSHS $\tilde{\mathscr{H}}$ defined with respect to a Borel safe set $W \in \mathscr{B}(S)$:

1. Compute the maximal reachability probability

$$\tilde{p}^*(\xi(p_0); \{1\} \times W) := \sup_{\tilde{\pi}' \in \tilde{\Pi}'} \tilde{p}^{\tilde{\pi}'}(\xi(p_0); \{1\} \times W);$$

2. Find an optimal policy $\tilde{\pi}^* \in \tilde{\Pi}'$, if it exists, such that $\tilde{p}^*(\xi(p_0); \{1\} \times W) = \tilde{p}^{\tilde{\pi}^*}(\xi(p_0); \{1\} \times W)$. Otherwise, for a choice of $\varepsilon > 0$, find an ε-optimal policy $\tilde{\pi}^*_\varepsilon \in \tilde{\Pi}'$ satisfying

$$\tilde{p}^{\tilde{\pi}^*_\varepsilon}(\xi(p_0); \{1\} \times W) \geq \tilde{p}^*(\xi(p_0); \{1\} \times W) - \varepsilon.$$

By the form of the cost function in (5.10), the above problem can be viewed as a terminal cost problem for $\tilde{\mathscr{H}}$. In the following, we will establish the equivalence between Problem 5.1 and Problem 5.2.

Proposition 5.1. *Let $\mathscr{H} = (Q, n, C_a, Z, \nu_x, \nu_q, \nu_r, \zeta_0, \zeta)$ be a POdtSHS, and $W \in \mathscr{B}(S)$ be a Borel safe set. Let $\tilde{\mathscr{H}} = (\tilde{S}, C_a, Z, \tilde{\nu}, \tilde{\zeta}_0, \tilde{\zeta})$ be the corresponding augmented POdtSHS. Then for every $p_0 \in \mathscr{P}(S)$, we have*

$$p^*(p_0; W) = \tilde{p}^*(\xi(p_0); \{1\} \times W).$$

Proof. For every $p_0 \in \mathscr{P}(S)$, a policy $\pi' \in \Pi'$ for $\mathscr{H}$ is equivalent to a policy $\tilde{\pi}' \in \tilde{\Pi}'$ for $\tilde{\mathscr{H}}$ by (5.9). Through this equivalence (as induced by the Borel isomorphism ξ), it is sufficient to prove that, for every $\pi' \in \Pi'$, the following equality holds:

$$p^{\pi'}(p_0; W) = \tilde{p}^{\pi'}(\xi(p); \{1\} \times W).$$

Indeed, by the previous definitions,

$$\begin{aligned}
\tilde{p}^{\pi'}(\xi(p_0);\{1\}\times W) &= \int_{\tilde{\Omega}_N} \mathbf{1}_{\{1\}\times W}(\tilde{s}_N)\tilde{\zeta}(dz_N|\tilde{s}_N,a_{N-1})\tilde{\nu}(d\tilde{s}_N|\tilde{s}_{N-1},a_{N-1}) \\
&\times \pi'_{N-1}(da_{N-1}|p_0;i_{N-1})\tilde{\zeta}(dz_{N-1}|\tilde{s}_{N-1},a_{N-2})\tilde{\nu}(d\tilde{s}_{N-1}|\tilde{s}_{N-2},a_{N-2}) \\
&\times \cdots \pi'_0(da_0|p_0;z_0)\tilde{\zeta}_0(dz_0|\tilde{s}_0)\xi(p_0)(d\tilde{s}_0) \\
&= \int_{\tilde{S}^N\times S\times Z^{N+1}\times C_a^N} \mathbf{1}_W(s_N)\zeta(dz_N|s_N,a_{N-1})\nu(ds_N|s_{N-1},a_{N-1}) \\
&\times \pi'_{N-1}(da_{N-1}|p_0;i_{N-1})\zeta(dz_{N-1}|s_{N-1},a_{N-2}) \\
&\times \mathbf{1}_{\{1\}\times W}(\tilde{s}_{N-1})\tilde{\nu}(d\tilde{s}_{N-1}|\tilde{s}_{N-2},a_{N-2})\pi'_{N-2}(da_{N-2}|p_0;i_{N-2}) \\
&\times \cdots \pi'_0(da_0|p_0;z_0)\zeta_0(dz_0|s_0)\xi(p_0)(d\tilde{s}_0) \\
&= \int_{\tilde{S}\times S^N\times Z^{N+1}\times C_a^N} \prod_{k=1}^{N} \mathbf{1}_W(s_k)\zeta(dz_N|s_N,a_{N-1})\nu(ds_N|s_{N-1},a_{N-1}) \\
&\times \pi'_{N-1}(da_{N-1}|p_0;i_{N-1})\zeta(dz_{N-1}|s_{N-1},a_{N-2})\nu(ds_{N-1}|s_{N-2},a_{N-2}) \\
&\times \cdots \times \pi'_0(da_0|p_0;z_0)\zeta_0(dz_0|s_0)\mathbf{1}_{\{1\}\times W}(\tilde{s}_0)\xi(p_0)(d\tilde{s}_0) \\
&= \int_{\Omega_N} \prod_{k=0}^{N} \mathbf{1}_W(s_k)dP_N(\pi',p_0) = p^{\pi'}(p_0;W).
\end{aligned}$$

This completes the proof. □

5.3.2 Construction of a Sufficient Statistic

By Proposition 5.1, the partial information safety problem for the original hybrid system $\mathscr{H}$ is equivalent to a terminal reachability problem for the augmented hybrid system $\tilde{\mathscr{H}}$. This allows us to use results from Bertsekas and Shreve (1978) to construct a statistic sufficient for control of the augmented system.

In the following, we adapt the definition given in Bertsekas and Shreve (1978) of a sufficient statistic for general additive cost stochastic optimal control problems to the terminal cost problem for $\mathscr{H}$.

Definition 5.4. A statistic for $\tilde{\mathscr{H}}$ is a sequence $(\eta_0,\eta_1,...\eta_{N-1})$ of Borel-measurable functions $\eta_k : \xi(\mathscr{P}(S))\times I_k \to B_k$, where $B_0,...,B_{N-1}$ are nonempty Borel spaces. A statistic $(\eta_0,\eta_1,...\eta_{N-1})$ for $\mathscr{H}$ is said to be sufficient for control if

1. For every $k=0,1,...,N-1$, there exists a Borel-measurable stochastic kernel $\hat{\nu}(d\eta_{k+1}|\eta_k,a)$ on B_{k+1} given $B_k\times C_a$ such that for every $p_0\in\mathscr{P}(S)$, $\tilde{\pi}'\in\tilde{\Pi}'$, and $E_{k+1}\in\mathscr{B}(B_{k+1})$, the following identity holds

$$\tilde{P}_{k+1}(\tilde{\pi}',\xi(p_0))\{\eta_{k+1}(\xi(p_0);i_{k+1})\in E_{k+1}|\eta_k(\xi(p_0);i_k)=\eta,a_k=a\} = \hat{\nu}(E_{k+1}|\eta,a)$$

for $\tilde{P}_k(\tilde{\pi}',\xi(p))$ almost every (η,a).

2. There exists a lower semianalytic function $g_N : B_N \to [0,1]$ such that for every $p_0 \in \mathscr{P}(S)$ and $\tilde{\pi}' \in \tilde{\Pi}'$, the following identity holds

$$E^{\tilde{\pi}'}_{\xi(p_0)}\left[\mathbf{1}_{\{1\}\times W}(\tilde{s}_N)|\eta_N(\xi(p_0);i_N)=\eta\right] = g_N(\eta)$$

for $\tilde{P}_k(\tilde{\pi}',\xi(p_0))$ almost every η.

For the derivation of a sufficient statistic, we will use the notations $\xi(p_0) \in \xi(\mathscr{P}(S))$ and $\tilde{p} \in \mathscr{P}(\tilde{S})$ to distinguish between an initial distribution in the range of ξ and a probability distribution on $\tilde{S}$ produced using recursive update equations. In particular, $\tilde{p}$ may not belong to the range of ξ.

As a first step, by Lemma 10.3 of Bertsekas and Shreve (1978), there exist Borel-measurable stochastic kernels $\Phi_0(d\tilde{s}|\xi(p);z)$ on $\tilde{S}$ given $\xi(\mathscr{P}(S)) \times Z$ and $\Phi(d\tilde{s}|\tilde{p};z,a)$ on $\tilde{S}$ given $\mathscr{P}(\tilde{S}) \times Z \times C_a$ which satisfy

$$\int_{E_1} \tilde{\xi}_0(E_2|\tilde{s})\xi(p_0)(d\tilde{s}) = \int_{\tilde{S}}\int_{E_2} \Phi_0(E_1|\xi(p_0);z)\tilde{\xi}_0(dz|\tilde{s})\xi(p_0)(d\tilde{s}) \tag{5.11}$$

$$\int_{E_1} \tilde{\xi}(E_2|\tilde{s},a)\tilde{p}(d\tilde{s}) = \int_{\tilde{S}}\int_{E_2} \Phi(E_1|\tilde{p};z,a)\tilde{\xi}(dz|\tilde{s},a)\tilde{p}(d\tilde{s}) \tag{5.12}$$

for every Borel set $E_1 \in \mathscr{B}(\tilde{S})$, $E_2 \in \mathscr{B}(Z)$, probability distribution $\xi(p_0) \in \xi(\mathscr{P}(S))$, $\tilde{p} \in \mathscr{P}(\tilde{S})$, and control action $a \in C_a$.

Now consider a function $\Psi : \mathscr{P}(\tilde{S}) \times C_a \to \mathscr{P}(\tilde{S})$, corresponding to the prediction step of a hybrid state filter:

$$\Psi(\tilde{p},a)(E) = \int_{\tilde{S}} \tilde{\nu}(E|\tilde{s},a)\tilde{p}(d\tilde{s}),\ \forall E \in \mathscr{B}(\tilde{S}). \tag{5.13}$$

By Propositions 7.26 and 7.29 of Bertsekas and Shreve (1978), the mapping Ψ is Borel-measurable. For a given information vector $i_k \in I_k$ and initial distribution $\xi(p_0) \in \xi(\mathscr{P}(S))$, define the stochastic kernels $\tilde{p}_k : \xi(\mathscr{P}(S)) \times I_k \to \mathscr{P}(\tilde{S})$ through the following innovation equations:

$$\begin{aligned} \tilde{p}_0(\xi(p_0);i_0) &= \Phi_0(d\tilde{s}_0|\xi(p_0);z_0), \\ \tilde{p}_{k+1}(\xi(p_0);i_{k+1}) &= \Phi(d\tilde{s}_{k+1}|\Psi(\tilde{p}_k(\xi(p_0);i_k),a_k);z_{k+1},a_k). \end{aligned} \tag{5.14}$$

Clearly, these stochastic kernels are Borel-measurable. Furthermore, by Lemma 10.4 of Bertsekas and Shreve (1978), $\tilde{p}_k(\xi(p_0);i_k)$ can be viewed as the conditional distribution of (h_k,s_k) given the information vector i_k and initial distribution $\xi(p_0)$. Finally, by Proposition 10.5 of Bertsekas and Shreve (1978), we have that the sequence $\{\tilde{p}_k(\xi(p_0);i_k)\}_{k=0}^{N-1}$ is a sufficient statistic for $\tilde{\mathscr{H}}$.

In particular, a transition kernel $\hat{\nu}$ for the statistic $\tilde{p}_k$ can be defined as

$$\hat{\nu}(E_{k+1}|\tilde{p}_k,a_k) = \int_{\tilde{S}}\int_{\tilde{S}} \zeta(G(\tilde{p}_k,a_k,E_{k+1})|\tilde{s}_{k+1},a_k)\tilde{\nu}(d\tilde{s}_{k+1}|\tilde{s}_k,a_k)\tilde{p}_k(d\tilde{s}_k), \tag{5.15}$$

where $G(\tilde{p},a,E) = \{z \in Z|\Phi(\cdot|\Psi(\tilde{p},a);z,a) \in E\}$. Thus, the evolution of $\tilde{p}_k(\xi(p);i_k)$ can be characterized exclusively in terms of the stochastic kernel $\hat{\nu}$. For the rest of this paper, we will refer to

$\tilde{p}_k$ as an *information state*. The terminal cost with respect to this information state can be defined as

$$g_N(\tilde{p}_N) := \int_{\tilde{S}} \mathbf{1}_{\{1\}\times W}(\tilde{s}_N)\tilde{p}_N(d\tilde{s}_N) = \int_W \tilde{p}_N(1, ds_N). \tag{5.16}$$

In the following section, this function will be used to construct an equivalent perfect information stochastic optimal control problem on the space of information states.

5.3.3 Reduction to Perfect State Information Problem

Consider a perfect state information model $\hat{\mathscr{H}}$ in which the state space is given by $\hat{S} := \mathscr{P}(\tilde{S})$, the action space is given by C_a, and the state transition kernel is given by $\hat{\nu}$. Define the set of admissible control policies for $\hat{\mathscr{H}}$ as follows.

Definition 5.5. A policy $\hat{\pi}'$ for $\hat{\mathscr{H}}$ is a sequence $\hat{\pi} = (\hat{\pi}'_0, \hat{\pi}'_1, ..., \hat{\pi}'_{N-1})$ of universally measurable stochastic kernels $\hat{\pi}'_k : \mathscr{B}(C_a) \times \hat{S}^{k+1} \times C_a^k \to [0,1]$, assigning to each sequence of controls and information states $(\tilde{p}_0, a_0, ..., \tilde{p}_{k-1}, a_{k-1}, \tilde{p}_k)$ a probability measure $\hat{\pi}'_k(da_k|\tilde{p}_0, a_0, ..., \tilde{p}_{k-1}, a_{k-1}, \tilde{p}_k)$ on the Borel space $(C_a, \mathscr{B}(C_a))$. The set of such policies is denoted by $\hat{\Pi}'$.

If for each k, the stochastic kernel $\hat{\pi}'_k$ depends on the history only through the current information state $\tilde{p}_k$, then the policy $\hat{\pi}'$ is said to be *Markov*. If for each k and history vector $(\tilde{p}_0, a_0, ..., \tilde{p}_{k-1}, a_{k-1}, \tilde{p}_k)$, the stochastic kernel $\hat{\pi}'_k$ assigns probability mass one to some point in C_a, the policy $\hat{\pi}'$ is said to be *non-randomized*. The class of non-randomized, Markov policies for $\hat{\mathscr{H}}$ is denoted as $\hat{\Pi}$. For any $\hat{\pi}' \in \hat{\Pi}$, we can identify the stochastic kernels $\hat{\pi}'_k$ with a sequence of universally measurable maps $\hat{\pi}_k : \hat{S} \to C_a$ (see for example Bertsekas and Shreve, 1978, Corollary 7.44.3).

Let $\hat{\pi}' \in \hat{\Pi}'$, then by Proposition 7.44 of Bertsekas and Shreve (1978), the sequence $\tilde{\pi}' = (\tilde{\pi}'_0, \tilde{\pi}'_1, ..., \tilde{\pi}'_{N-1})$ defined by

$$\tilde{\pi}'_k(da_k|\xi(p_0); i_k) = \hat{\pi}'_k(da_k|\tilde{p}_0(\xi(p_0); i_0), a_0, ..., a_{k-1}, \tilde{p}_k(\xi(p_0); i_k)) \tag{5.17}$$

is a policy belonging to $\tilde{\Pi}'$. Through this identification, we can view $\hat{\Pi}'$ as a subset of $\tilde{\Pi}'$, and hence also of Π'.

Now consider the sample space of information state and control sequences over time horizon N given by $\hat{\Omega}_N := \hat{S}^{N+1} \times C_a^N$, equipped with the canonical product topology $\mathscr{B}(\hat{\Omega}_N) := \prod_{k=1}^{N+1} \mathscr{B}(\hat{S}) \times \prod_{k=1}^{N} \mathscr{B}(C_a)$. Then for a given initial information state $\tilde{p}_0 \in \hat{S}$ and policy $\hat{\pi}' \in \hat{\Pi}'$, the stochastic kernels $\hat{\nu}$ and $\hat{\pi}'_k$, $k = 0, 1, .., N$ induce a unique probability measure $\hat{P}^{\hat{\pi}'}_{\tilde{p}_0}$ on $\hat{\Omega}_N$.

Let $\tilde{p}_0 \in \hat{S}$, $\hat{\pi}' \in \hat{\Pi}'$, consider an N-stage cost function defined by

$$J_{N,\hat{\pi}'}(\tilde{p}_0) := \int_{\hat{\Omega}_N} g_N(\tilde{p}_N) d\hat{P}^{\hat{\pi}'}_{\tilde{p}_0}. \tag{5.18}$$

The perfect state information problem for $\hat{\mathscr{H}}$ is stated as follows.

Problem 5.3. Given a perfect state information model $\hat{\mathscr{H}}$ defined with respect to a Borel safe set $W \in \mathscr{B}(S)$:

1. Compute the optimal cost $J_N^* := \sup_{\hat{\pi}' \in \hat{\Pi}'} J_{N,\hat{\pi}'}$;
2. Find an optimal policy $\hat{\pi}^* \in \hat{\Pi}'$, if it exists, such that $J_N^*(\tilde{p}_0) = J_{N,\hat{\pi}^*}(\tilde{p}_0)$, $\forall \tilde{p}_0 \in \hat{S}$. Otherwise, for a choice of $\varepsilon > 0$, find an ε-optimal policy $\hat{\pi}_\varepsilon^* \in \hat{\Pi}'$ satisfying

$$J_{N,\hat{\pi}_\varepsilon^*}(\tilde{p}_0) \geq J_N^*(\tilde{p}_0) - \varepsilon, \ \forall \tilde{p}_0 \in \hat{S}.$$

Given the terminal cost structure in (5.18), we can apply standard dynamic programming results for additive cost problems to obtain a solution to Problem 5.3 (see for example Bertsekas and Shreve, 1978, chapter 8). In particular, the ε-optimal policies can be found within the class of non-randomized Markov policies $\hat{\Pi}$. Before the discussion of this dynamic programming solution, we will first establish the connection between Problem 5.1 and Problem 5.3.

Proposition 5.2. *Let $\mathscr{H} = (Q, n, C_a, Z, \nu_x, \nu_q, \nu_r, \zeta_0, \zeta)$ be a POdtSHS and $W \in \mathscr{B}(S)$ be a Borel safe set. Let $\hat{\mathscr{H}} = (\hat{S}, C_a, \hat{\nu})$ be the corresponding perfect state information model. Define a function $\varphi : \mathscr{P}(S) \to \mathscr{P}(\hat{S})$ as*

$$\varphi(p_0)(E) = \int_S \zeta_0(\{z_0 | \tilde{p}_0(\xi(p_0); z_0) \in E\} | s_0) p_0(ds_0), \tag{5.19}$$

for every Borel set $E \in \mathscr{B}(\hat{S})$. Then we have

$$p^*(p_0; W) = \int_{\hat{S}} J_N^*(\tilde{p}_0) \varphi(p_0)(d\tilde{p}_0), \ \forall p_0 \in \mathscr{P}(S).$$

Furthermore, if $\hat{\pi}' \in \hat{\Pi}'$ is optimal, or ε-optimal for Problem 5.3, then $\hat{\pi}'$ is also optimal, or ε-optimal for Problem 5.1.

Proof. Let $\tilde{\varphi} : \xi(\mathscr{P}(S)) \to \mathscr{P}(\hat{S})$ be defined as

$$\tilde{\varphi}(\xi(p_0))(E) = \int_{\tilde{S}} \tilde{\zeta}_0(\{z_0 | \tilde{p}_0(\xi(p_0); z_0) \in E\} | \tilde{s}_0) \xi(p_0)(d\tilde{s}_0),$$

for every Borel set $E \in \mathscr{B}(S)$. Then it follows by Proposition 10.3 of Bertsekas and Shreve (1978) that

$$\tilde{p}^*(\xi(p_0); \{1\} \times W) = \int_{\hat{S}} J_N^*(\tilde{p}_0) \tilde{\varphi}(\xi(p_0))(d\tilde{p}_0),$$

for every $\xi(p_0) \in \xi(\mathscr{P}(S))$. Furthermore, if $\hat{\pi}' \in \hat{\Pi}'$ is optimal, or ε-optimal for Problem 5.3, then $\hat{\pi}'$ is also optimal, or ε-optimal for Problem 5.2.

Thus, by Proposition 5.1 and the observation that $\varphi(p_0) = \tilde{\varphi}(\xi(p_0))$, $\forall p_0 \in \mathscr{P}(S)$, we have the desired conclusion. □

5.4 Solution to Partial Information Safety Problem

As shown in Proposition 5.2, solving a partial information safety problem defined on the hybrid state space is equivalent to solving a perfect information terminal cost problem defined on the information state space. In this section, we will first focus on solving Problem 5.3. This then in turn provides a solution to Problem 5.1.

Specifically, consider a dynamic programming operator $\mathscr{T}_{Safe}$, which takes as its argument an universally measurable function $J : \hat{S} \to [0,1]$ and returns a function $\mathscr{T}_{Safe}(J) : \hat{S} \to [0,1]$:

$$\mathscr{T}_{Safe}(J)(\tilde{p}) = \sup_{a \in C_a} \int_{\hat{S}} J(\tilde{p}')\hat{\nu}(d\tilde{p}'|\tilde{p},a), \ \tilde{p} \in \hat{S}. \tag{5.20}$$

The solution to Problem 5.3 is given as follows.

Proposition 5.3. *Let $\hat{\mathscr{H}} = (\hat{S}, C_a, \hat{\nu})$ be a perfect state information model defined with respect to a Borel safe set $W \in \mathscr{B}(S)$. Then*

1. $J_N^* = \mathscr{T}_{Safe}^N(g_N)$;
2. *For every $\varepsilon > 0$, there exists an ε-optimal non-randomized Markov policy $\hat{\pi}_\varepsilon^* \in \hat{\Pi}$ for Problem 5.3. In particular,*

$$J_N^* := \sup_{\hat{\pi}' \in \hat{\Pi}'} J_{N,\hat{\pi}'} = \sup_{\hat{\pi} \in \hat{\Pi}} J_{N,\hat{\pi}}.$$

Proof. These statements are direct consequences of Propositions 8.2, 8.3, and 10.1 of Bertsekas and Shreve (1978). □

From this result, we have that J_N^* can be computed through recursive applications of the dynamic programming operator $\mathscr{T}_{Safe}$, initialized with the terminal cost g_N, and that the set of non-randomized Markov policies $\hat{\Pi}$ is optimal over the set of general policies $\hat{\Pi}'$. Furthermore, sufficient conditions of optimality can be also derived from the dynamic programming algorithm. For notational conveniences, we define the optimal cost-to-go functions $J_{k\to N}^* : \hat{S} \to [0,1]$ by

$$J_{k\to N}^* := \mathscr{T}_{Safe}^{N-k}(g_N), \ k = 0,1,...,N. \tag{5.21}$$

By Proposition 5.3, it follows that $J_{0\to N}^* = J_N^*$. Using this fact and standard dynamic programming arguments (see for example Bertsekas and Shreve, 1978, Proposition 8.2), we obtain the following corollary.

Corollary 5.1.

1. *If $\hat{\pi} = (\hat{\pi}_0, \hat{\pi}_1, ..., \hat{\pi}_{N-1}) \in \hat{\Pi}$ satisfies*

$$\hat{\pi}_k(\tilde{p}) \in \arg\sup_{a \in C_a} \int_{\hat{S}} J_{k+1\to N}^*(\tilde{p}')\hat{\nu}(d\tilde{p}'|\tilde{p},a), \tag{5.22}$$

for every $\tilde{p} \in \hat{S}$ and $k = 0,1...,N-1$, then $\hat{\pi}$ is an optimal policy for Problem 5.3.

2. *For a given $\varepsilon > 0$, let $\{\varepsilon_k\}_{k=0}^{N-1}$ be any sequence of positive real numbers such that $\sum_{k=0}^{N-1} \varepsilon_k = \varepsilon$. If $\hat{\pi} = (\hat{\pi}_0, \hat{\pi}_1, ..., \hat{\pi}_{N-1}) \in \hat{\Pi}$ satisfies*

$$\int_{\hat{S}} J^*_{k+1\to N}(\tilde{p}')\hat{\nu}(d\tilde{p}'|\tilde{p}, \hat{\pi}_k(\tilde{p})) \geq J^*_{k\to N}(\tilde{p}) - \varepsilon_k, \tag{5.23}$$

for every $\tilde{p} \in \hat{S}$ and $k = 0, 1..., N-1$, then $\hat{\pi}$ is an ε-optimal policy for Problem 5.3.

Combining Propositions 5.2 and 5.3, and Corollary 5.1, we arrive at the main result of this chapter.

Theorem 5.1. *Let $\mathscr{H}$ be a POdtSHS and $W \in \mathscr{B}(S)$ be a Borel safe set. Let $\hat{\mathscr{H}}$ be the corresponding perfect state information model. Define $g_N : \hat{S} \to [0,1]$ as in (5.16) and $\varphi : \mathscr{P}(S) \to \mathscr{P}(\hat{S})$ as in (5.19). Then given $p_0 \in \mathscr{P}(S)$, we have*

1. *$p^*(p_0; W) = \int_{\hat{S}} \mathscr{T}^N_{Safe}(g_N)(\tilde{p}_0)\varphi(p_0)(d\tilde{p}_0)$;*
2. *For every $\varepsilon > 0$, there exists an ε-optimal non-randomized policy $\pi^*_\varepsilon \in \Pi$ for Problem 5.1 of the form*

$$\pi^*_{k,\varepsilon}(p_0; i_k) = \hat{\pi}_{k,\varepsilon}(\tilde{p}_k(\xi(p_0); i_k)),\ k = 0, 1, ..., N-1.$$

In particular,

$$p^*(p_0; W) := \sup_{\pi' \in \Pi'} p^{\pi'}(p_0; W) = \sup_{\hat{\pi} \in \hat{\Pi}} p^{\hat{\pi}}(p_0; W).$$

3. *If $\hat{\pi} = (\hat{\pi}_0, \hat{\pi}_1, ..., \hat{\pi}_{N-1}) \in \hat{\Pi}$ satisfies (5.22), then $\hat{\pi}$ is an optimal policy for Problem 5.1. For a given $\varepsilon > 0$, if $\hat{\pi} = (\hat{\pi}_0, \hat{\pi}_1, ..., \hat{\pi}_{N-1}) \in \hat{\Pi}$ satisfies (5.23), then $\hat{\pi}$ is an ε-optimal policy for Problem 5.1.*

By this result, the optimal safety probability $p^*(p_0; W)$ for the POdtSHS can be computed through a terminal cost dynamic programming algorithm on the information state space $\hat{S}$. Furthermore, the ε-optimal policies can be found within the class of non-randomized policies which depends on the initial distribution p_0 and observation history i_k only through the augmented information state $\tilde{p}_k(\xi(p_0); i_k)$. This decouples the partial information safety problem into two subproblems:

1. Computing the ε-optimal control policy $\hat{\pi}^*_\varepsilon$ via the dynamic programming recursion given in Proposition 5.3;
2. Computing the conditional state distribution $\tilde{p}_k(\xi(p_0); i_k)$ through the innovation equations (5.14).

The first subproblem, which is the *control* aspect of the problem, can be performed in an offline setting, while the second subproblem, which is the *estimation* aspect of the problem, has to be performed in an online setting. In both of these problems, the difficulty of finding computationally tractable solutions is to a large extent associated with the representation of the information state $\tilde{p}_k$,

as it determines the size of the space in which the filtering and dynamic programming algorithms need to take place. Given that $\tilde{p}_k$ is a probability distribution, which is infinite dimensional, rather than a hybrid state s_k, which is finite dimensional, it can be seen that the partial information safety problem is in general significantly more difficult than its perfect information counterpart.

It is observed in Kumar and Varaiya (1986) that the abrupt jump in complexity when one moves from a perfect information model to a partial information model is reflective of the dual role of control in a partial information optimal control problem. Namely the choice of control affects not only the evolution of the actual state through the system dynamics, but also the sequence of observations generated by the state trajectory, and hence the availability of information. If more informative measurements or observations are made, then the uncertainty in state estimates would likely reduce, leading to better choices of control inputs. However, the payoff gained by having better estimates needs to be balanced with the payoff lost in the process of obtaining better estimates. The use of an information state as a characterization of the estimation uncertainty allows the control to quantify the expected costs and benefits of a reduction in uncertainty. This improvement in control quality unfortunately comes at the expense of reasoning on the space of uncertainty representations, which may be much larger than the underlying hybrid state space.

For cases in which it is possible to find an ε-optimal control policy $\hat{\pi}^*_\varepsilon$ for Problem 5.3, a control algorithm for the POdtSHS can be implemented, at least in principle, according to Algorithm 5.4.1. A block diagram illustration of this algorithm is shown in Figure 5.2.

Algorithm 5.4.1 POdtSHS Control Algorithm

Require: Initial distribution $p_0 \in \mathscr{P}(S)$ and policy $\hat{\pi}^*_\varepsilon \in \hat{\Pi}$.
  for $k = 0$ to $N-1$ **do**
    Obtain a measurement z_k;
    Compute information state $\tilde{p}_k(\xi(p_0); i_k)$ using (5.14);
    Apply control input $a_k = \hat{\pi}_{k,\varepsilon}(\tilde{p}_k(\xi(p_0); i_k))$;
  end for

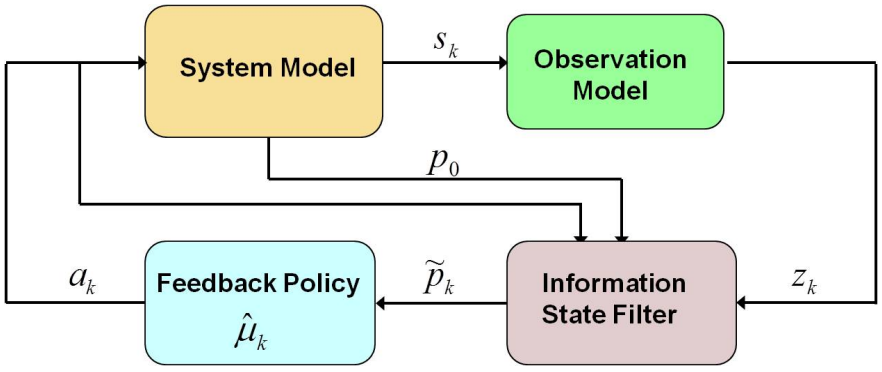


Figure 5.2: Block diagram of POdtSHS control algorithm.

5.5 Extension to Probabilistic Reach-avoid Problem

In this section, we will discuss how the analysis of the preceding sections can be extended to address the reach-avoid problem. In particular, due to the sum-multiplicative cost structure, the reach-avoid problem is equivalent to an additive cost stochastic optimal control problem.

More specifically, suppose that $R \in \mathscr{B}(S)$ is given as the target set and $W' \in \mathscr{B}(S)$ as the safe set, with $R \subseteq W'$. Then the probability that the state trajectory $(s_0, s_1, ..., s_N)$ of a POdtSHS $\mathscr{H}$ reaches R while staying inside W' for an initial distribution $p_0 \in \mathscr{P}(W')$ and $\pi' \in \Pi'$ is given by

$$\begin{aligned}
&r^{\pi'}(p_0; R, W') \\
&:= P_N(\pi', p_0)(\{(s_0, z_0, a_0, ..., s_N, z_N) : \exists k \in [0,N], (s_k \in R) \wedge (s_j \in W',\ \forall j \in [0,k])\}) \\
&= P_N(\pi', p_0)\left(\bigcup_{k=0}^{N} (W' \setminus R)^k \times R \times S^{N-k} \times Z^{N+1} \times C_a^N\right) \\
&= \sum_{k=0}^{N} P_N(\pi', p_0)((W' \setminus R)^k \times R \times S^{N-k} \times Z^{N+1} \times C_a^N), \qquad (5.24)
\end{aligned}$$

where the final equality follows by the fact that the union is disjoint. From (5.1), this probability can be computed as

$$r^{\pi'}(p_0; R, W') = E_{p_0}^{\pi'}\left[\sum_{k=0}^{N}\left(\prod_{j=0}^{k-1} \mathbf{1}_{W' \setminus R}(s_j)\right)\mathbf{1}_R(s_k)\right], \qquad (5.25)$$

where $E_{p_0}^{\pi'}$ denotes the expectation with respect to the probability measure $P_N(\pi', p_0)$ on the sample space Ω_N. As before, our control objective is to maximize this probability over the general policy space Π'. More precisely, the partial information reach-avoid problem for a POdtSHS is as follows:

Problem 5.4. Given a POdtSHS $\mathscr{H}$, initial distribution $p_0 \in \mathscr{P}(S)$, target set $R \in \mathscr{B}(S)$, and safe set $W' \in \mathscr{B}(S)$ such that $R \subseteq W'$:

1. Compute the maximal reach-avoid probability

$$r^*(p_0; R, W') := \sup_{\pi' \in \Pi'} r^{\pi'}(p_0; R, W');$$

2. Find an optimal policy $\pi^* \in \Pi'$, if it exists, such that $r^*(p_0; R, W') = r^{\pi^*}(p_0; R, W')$. Otherwise, for a choice of $\varepsilon > 0$, find an ε-optimal policy $\pi_\varepsilon^* \in \Pi'$ satisfying

$$r^{\pi_\varepsilon^*}(p_0; R, W') \geq r^*(p_0; R, W') - \varepsilon.$$

We define a modified history state for the reach-avoid problem as

$$h_0 = 1;\ h_k = \prod_{j=0}^{k-1} \mathbf{1}_{W' \setminus R}(s_j),\ k \geq 1. \qquad (5.26)$$

The corresponding augmented POdtSHS model $\tilde{\mathscr{H}}$, whose state at each time step k is given by (h_k, s_k), can be defined similarly as in Section 5.3.1.

For a given initial distribution $\xi(p_0) \in \xi(\mathscr{P}(S))$ and policy $\tilde{\pi}' \in \tilde{\Pi}'$ for $\tilde{\mathscr{H}}$, consider the following additive cost function

$$\tilde{r}^{\tilde{\pi}'}(\xi(p_0); R, W') := E^{\tilde{\pi}'}_{\xi(p_0)}\left[\sum_{j=0}^{N} \mathbf{1}_{\{1\}\times R}(\tilde{s}_j)\right], \tag{5.27}$$

where $E^{\tilde{\pi}'}_{\xi(p_0)}$ denotes the expectation with respect to the probability measure $\tilde{P}_N(\tilde{\pi}', \xi(p_0))$ on the sample space $\tilde{\Omega}_N := \tilde{S}^{N+1} \times Z^{N+1} \times C_a^N$. In the following, we use $\tilde{r}^{\tilde{\pi}'}(\xi(p_0); R, W')$ to define an additive cost optimal control problem for the augmented POdtSHS.

Problem 5.5. Given an initial distribution $p_0 \in \mathscr{P}(S)$ and an augmented POdtSHS $\tilde{\mathscr{H}}$ defined with respect to a target set $R \in \mathscr{B}(S)$ and a safe set $W' \in \mathscr{B}(S)$ such that $R \subseteq W'$:

1. Compute the optimal cost

$$\tilde{r}^*(\xi(p_0); R, W') := \sup_{\tilde{\pi}' \in \tilde{\Pi}'} \tilde{r}^{\tilde{\pi}'}(\xi(p_0); R, W');$$

2. Find an optimal policy $\tilde{\pi}^* \in \tilde{\Pi}'$, if it exists, such that $\tilde{r}^*(\xi(p_0); R, W') = \tilde{r}^{\tilde{\pi}^*}(\xi(p_0); R, W')$. Otherwise, for a choice of $\varepsilon > 0$, find an ε-optimal policy $\tilde{\pi}^*_\varepsilon \in \tilde{\Pi}'$ satisfying

$$\tilde{r}^{\tilde{\pi}^*_\varepsilon}(\xi(p_0); R, W') \geq \tilde{r}^*(\xi(p_0); R, W') - \varepsilon.$$

We now proceed to establish the equivalence between Problem 5.4 and Problem 5.5.

Proposition 5.4. *For every $p_0 \in \mathscr{P}(S)$ and $R, W' \in \mathscr{B}(S)$ such that $R \subseteq W'$, we have*

$$r^*(p_0; R, W') = \tilde{r}^*(\xi(p_0); R, W').$$

Proof. By the equivalence of the policy spaces Π' and $\tilde{\Pi}'$, it is again sufficient to prove that, for every $\pi' \in \Pi'$, the following equality holds:

$$r^{\pi'}(p_0; R, W') = \tilde{r}^{\pi'}(\xi(p_0); R, W').$$

By the previous definitions,

$$\begin{aligned}\tilde{r}^{\pi'}(\xi(p_0); R, W') &= \int_{\tilde{\Omega}_N} \sum_{k=0}^{N} \mathbf{1}_{\{1\}\times R}(\tilde{s}_k) d\tilde{P}_N(\pi', \xi(p_0)) \\ &= \sum_{k=0}^{N} \int_{\tilde{\Omega}_k} \mathbf{1}_{\{1\}\times R}(\tilde{s}_k) d\tilde{P}_k(\pi', \xi(p_0))\end{aligned}$$

Using a similar line of argument as in the proof of Proposition 5.1, it can be shown that for every $k = 0, 1, ..., N$,

$$\int_{\tilde{\Omega}_k} \mathbf{1}_{\{1\}\times R}(\tilde{s}_k) d\tilde{P}_k(\pi', \xi(p_0)) = \int_{\Omega_k} \left(\prod_{j=0}^{k-1} \mathbf{1}_{W'\setminus R}(s_j) \right) \mathbf{1}_R(s_k) dP_k(\pi', p_0).$$

Thus, we have

$$\begin{aligned}
\tilde{r}^{\pi'}(\xi(p_0); R, W') &= \sum_{k=0}^{N} \int_{\Omega_k} \left(\prod_{j=0}^{k-1} \mathbf{1}_{W'\setminus R}(s_j) \right) \mathbf{1}_R(s_k) dP_k(\pi', p_0) \\
&= \int_{\Omega_N} \sum_{k=0}^{N} \left(\prod_{j=0}^{k-1} \mathbf{1}_{W'\setminus R}(s_j) \mathbf{1}_R(s_k) \right) dP_N(\pi', p_0) \\
&= E_{p_0}^{\pi'} \left[\sum_{k=0}^{N} \left(\prod_{j=0}^{k-1} \mathbf{1}_{W'\setminus R}(s_j) \right) \mathbf{1}_R(s_k) \right] = r^{\pi'}(p_0; R, W')
\end{aligned}$$

The desired conclusion then follows. □

Applying the set of procedures given in Section 5.3.2, we can derive an information state $\tilde{p}_k(\xi(p_0); i_k)$, $k = 0, 1, ..., N$ for the augmented POdtSHS model $\tilde{\mathscr{H}}$. By Proposition 10.5 of Bertsekas and Shreve (1978), this then becomes a sufficient statistic for Problem 5.5. Let $\mathscr{H} = (\hat{S}, C_a, \hat{\nu})$ be the corresponding perfect state information model. Consider a Borel-measurable one-stage cost $g : \hat{S} \to [0, 1]$ defined by

$$g(\tilde{p}) := \int_{\tilde{S}} \mathbf{1}_{\{1\}\times R}(\tilde{s}) \tilde{p}(d\tilde{s}) = \int_R \tilde{p}(1, ds), \tag{5.28}$$

Let $\tilde{p}_0 \in \hat{S}$, $\hat{\pi}' \in \hat{\Pi}'$, define an N-stage cost function as

$$\tilde{J}_{N,\hat{\pi}'}(\tilde{p}_0) := \int_{\hat{\Omega}_N} \sum_{j=0}^{N} g(\tilde{p}_j) d\hat{P}_{\tilde{p}_0}^{\hat{\pi}'}. \tag{5.29}$$

The perfect state information problem is stated as follows.

Problem 5.6. Given a perfect state information model $\hat{\mathscr{H}}$ defined with respect to a target set $R \in \mathscr{B}(S)$ and a safe set $W' \in \mathscr{B}(S)$ such that $R \subseteq W'$:

1. Compute the optimal cost $\tilde{J}_N^* := \sup_{\hat{\pi}' \in \hat{\Pi}'} \tilde{J}_{N,\hat{\pi}'}$;
2. Find an optimal policy $\hat{\pi}^* \in \hat{\Pi}'$, if it exists, such that $\tilde{J}_N^*(\tilde{p}_0) = \tilde{J}_{N,\hat{\pi}^*}(\tilde{p}_0)$, $\forall \tilde{p}_0 \in \hat{S}$. Otherwise, for a choice of $\varepsilon > 0$, find an ε-optimal policy $\hat{\pi}_\varepsilon^* \in \hat{\Pi}'$ satisfying

$$\tilde{J}_{N,\hat{\pi}_\varepsilon^*}(\tilde{p}_0) \geq \tilde{J}_N^*(\tilde{p}_0) - \varepsilon, \ \forall \tilde{p}_0 \in \hat{S}.$$

By an almost identical argument as in the proof of Proposition 5.2, we have the following result establishing the connection between Problem 5.4 and Problem 5.6.

Proposition 5.5. *Let $\mathscr{H} = (Q, n, C_a, Z, \nu_x, \nu_q, \nu_r, \zeta_0, \zeta)$ be a POdtSHS and $R, W' \in \mathscr{B}(S)$ be Borel sets such that $R \subseteq W'$. Let $\hat{\mathscr{H}} = (\hat{S}, C_a, \hat{\nu})$ be the corresponding perfect state information model for the reach-avoid problem. Define a function $\varphi : \mathscr{P}(S) \to \mathscr{P}(\hat{S})$ as in (5.19). Then we have*

$$r^*(p_0; R, W') = \int_{\hat{S}} \tilde{J}_N^*(\tilde{p}_0)\varphi(p_0)(d\tilde{p}_0), \ \forall p_0 \in \mathscr{P}(S).$$

Furthermore, if $\hat{\pi}' \in \hat{\Pi}'$ is optimal, or ε-optimal for Problem 5.6, then $\hat{\pi}'$ is also optimal, or ε-optimal for Problem 5.4.

Using standard dynamic programming results for additive cost problems, we can also derive a solution to Problem 5.6, which in turn provides a solution to Problem 5.4. Specifically, consider a dynamic programming operator $\mathscr{T}_{RA}$ as defined by

$$\mathscr{T}_{RA}(J)(\tilde{p}) = \sup_{a \in C_a} g(\tilde{p}) + \int_{\hat{S}} J(\tilde{p}')\hat{\nu}(d\tilde{p}'|\tilde{p}, a), \ \tilde{p} \in \hat{S} \tag{5.30}$$

for universally measurable functions $J : \hat{S} \to [0, 1]$.

Then by propositions 8.2 and 8.3 of Bertsekas and Shreve (1978), we have the following dynamic programming result.

Proposition 5.6. *Let $\hat{\mathscr{H}} = (\hat{S}, C_a, \hat{\nu})$ be a perfect state information model defined with respect to a target set $R \in \mathscr{B}(S)$ and a safe set $W' \in \mathscr{B}(S)$ such that $R \subseteq W'$. Then*

1. *$\tilde{J}_N^* = \mathscr{T}_{RA}^N(g)$;*
2. *For every $\varepsilon > 0$, there exists an ε-optimal non-randomized Markov policy $\hat{\pi}_\varepsilon^* \in \hat{\Pi}$ for Problem 5.6. In particular,*

$$\tilde{J}_N^* := \sup_{\hat{\pi}' \in \hat{\Pi}'} \tilde{J}_{N,\hat{\pi}'} = \sup_{\hat{\pi} \in \hat{\Pi}} \tilde{J}_{N,\hat{\pi}}.$$

Combining propositions 5.5 and 5.6, a solution to Problem 5.4 can be now stated.

Theorem 5.2. *Let $\mathscr{H}$ be a POdtSHS and $R, W' \in \mathscr{B}(S)$ be Borel sets such that $R \subseteq W'$. Let $\hat{\mathscr{H}}$ be the corresponding perfect state information model for the reach-avoid problem. Define $g : \hat{S} \to [0, 1]$ as in (5.28) and $\varphi : \mathscr{P}(S) \to \mathscr{P}(\hat{S})$ as in (5.19). Then given $p_0 \in \mathscr{P}(S)$, we have*

1. *$r^*(p_0; R, W') = \int_{\hat{S}} \mathscr{T}_{RA}^N(g)(\tilde{p}_0)\varphi(p_0)(d\tilde{p}_0)$;*
2. *For every $\varepsilon > 0$, there exists an ε-optimal non-randomized policy $\pi_\varepsilon^* \in \Pi$ for Problem 5.4 of the form*

$$\pi_{k,\varepsilon}^*(p_0; i_k) = \hat{\pi}_{k,\varepsilon}(\tilde{p}_k(\xi(p_0); i_k)), \ k = 0, 1, ..., N-1.$$

In particular,

$$r^*(p_0; R, W') := \sup_{\pi' \in \Pi'} r^{\pi'}(p_0; R, W') = \sup_{\hat{\pi} \in \hat{\Pi}} r^{\hat{\pi}}(p_0; R, W').$$

3. Let $\tilde{J}^*_{k\to N} := \mathscr{T}_{RA}^{N-k}(g)$, $k = 0,1,...,N$. *If* $\hat{\pi} = (\hat{\pi}_0, \hat{\pi}_1, ..., \hat{\pi}_{N-1}) \in \hat{\Pi}$ *satisfies*

$$\hat{\pi}_k(\tilde{p}) \in \arg\sup_{a\in C_a} \int_{\hat{S}} \tilde{J}^*_{k+1\to N}(\tilde{p}')\hat{\nu}(d\tilde{p}'|\tilde{p},a), \tag{5.31}$$

for every $\tilde{p} \in \hat{S}$ *and* $k = 0,1...,N-1$, *then* $\hat{\pi}$ *is an optimal policy for Problem 5.4. For a given* $\varepsilon > 0$, *let* $\{\varepsilon_k\}_{k=0}^{N-1}$ *be any sequence of positive real numbers such that* $\sum_{k=0}^{N-1} \varepsilon_k = \varepsilon$. *If* $\hat{\pi} = (\hat{\pi}_0, \hat{\pi}_1, ..., \hat{\pi}_{N-1}) \in \hat{\Pi}$ *satisfies*

$$\int_{\hat{S}} \tilde{J}^*_{k+1\to N}(\tilde{p}')\hat{\nu}(d\tilde{p}'|\tilde{p},\hat{\pi}_k(\tilde{p})) \geq \tilde{J}^*_{k\to N}(\tilde{p}) - \varepsilon_k, \tag{5.32}$$

for every $\tilde{p} \in \hat{S}$ *and* $k = 0,1...,N-1$, *then* $\hat{\pi}$ *is an* ε*-optimal policy for Problem 5.4.*

Thus, the partial information reach-avoid problem can be solved using an additive cost dynamic programming algorithm on the information state space $\hat{S}$. For the rest of this chapter, we will focus our attention on the probabilistic safety problem, with the understanding that any result proved for the safety problem can be generalized to the reach-avoid problem through the set of minor modifications in argument as described in this section.

5.6 Sufficiency of Non-Randomized Markov Policies for the Perfect Information Safety Problem

In this section, we will state a consequence of the preceding results for the special case in which the hybrid state is perfectly observed. Specifically, consider a PODTSHS model $\mathscr{H}$ with observation space $Z = S$ and observation model $\zeta_0(dz|s) = \zeta(dz|s,a) = \delta_s$. For the system $\mathscr{H}$, it will be shown that the class of non-randomized Markov policies, which select input deterministically based upon measurement the current state $z_k = s_k$, is optimal for the probabilistic safety problem, within the general class of randomized non-Markov policies. In other words, for the perfect information case, it is unnecessary to randomize one's choice of controls or maintain memory of the history of hybrid states and controls, despite the multiplicative cost structure of the safety problem. This provides formal justification for the restriction of attention to this class of policies in previous work by Amin et al. (2006), Abate et al. (2008), and Summers and Lygeros (2010) on perfect information probabilistic reachability problems for DTSHS.

As consistent with a perfect state information model, we assume that the initial condition $s_0 \in S$ of $\mathscr{H}$ is measured. This results in an initial distribution $p_0 = \delta_{s_0}$. The maximal safety probability, as a specialization of the definitions in section 5.2.2, is then parameterized only by the initial condition s_0, and will be thus denoted simply as $p^*_{s_0}(W)$. It can be verified that a sufficient statistic for the perfect information safety problem is given by the sequence of functions $\eta_k : \xi(\mathscr{P}(S)) \times$

$I_k \to \tilde{S}$ defined as

$$\eta_0(\xi(p_0); i_0) = (1, z_0),$$
$$\eta_k(\xi(p_0); i_k) = (\prod_{j=0}^{k-1} \mathbf{1}_W(z_j), z_k),\ k \geq 1.$$

By the definition of the history state h_k and the fact that $z_k = s_k$ in the perfect information case, it follows that $\eta_k(\xi(p_0); i_k) = (h_k, s_k)$, $\forall k \geq 0$. In other words, the information state for $\mathscr{H}$ is simply the augmented system state $\tilde{s}_k = (h_k, s_k)$, with the transition kernel as defined in (5.5), and the terminal cost $g_N(\tilde{s}_N) = \mathbf{1}_{\{1\}\times W}(\tilde{s}_N)$. Thus, by a special case of Theorem 5.1, we have

$$p^*_{s_0}(W) = \sup_{\hat{\pi}\in\hat{\Pi}} p^{\hat{\pi}}_{s_0}(W),\ \forall s_0 \in S, \tag{5.33}$$

where $\hat{\Pi}$ is the class of policies consisting of elements $\hat{\pi} = (\hat{\pi}_0, \hat{\pi}_1, ..., \hat{\pi}_{N-1})$, such that $\hat{\pi}_k : \tilde{S} \to C_a$ is a deterministic function of the augmented state. Now consider a subclass $\mathscr{M}$ of such policies which selects inputs independently of the history state h_k, namely $\hat{\pi}_k(0, s_k) = \hat{\pi}_k(1, s_k) = \mu_k(s_k)$ for some function $\mu_k : S \to C_a$. In the following, we will proceed to prove that it is sufficient to restrict one's attention to the policy class $\mathscr{M}$.

Proposition 5.7. *Let $\mathscr{H}$ be a perfect state information model. Then given an initial condition $s_0 \in S$ and a safe set $W \in \mathscr{B}(S)$, we have*

$$p^*_{s_0}(W) = \sup_{\mu\in\mathscr{M}} p^{\mu}_{s_0}(W),\ \forall s_0 \in S.$$

Proof. Given (5.33), it is sufficient to prove that the following equality holds:

$$\sup_{\hat{\pi}\in\hat{\Pi}} p^{\hat{\pi}}_{s_0}(W) = \sup_{\mu\in\mathscr{M}} p^{\mu}_{s_0}(W),\ \forall s_0 \in W.$$

First, by virtue of $\mathscr{M}$ being a subset of $\hat{\Pi}$, it can be inferred that

$$\sup_{\hat{\pi}\in\hat{\Pi}} p^{\hat{\pi}}_{s_0}(W) \geq \sup_{\mu\in\mathscr{M}} p^{\mu}_{s_0}(W),\ \forall s_0 \in S.$$

Now we proceed to show the reverse inequality. Fix any policy $\hat{\pi} = (\hat{\pi}_0, \hat{\pi}_1, ..., \hat{\pi}_{N-1}) \in \hat{\Pi}$, consider a policy $\mu = (\mu_0, \mu_1, ..., \mu_{N-1}) \in \mathscr{M}$ defined as

$$\mu_k(s_k) = \hat{\pi}_k(1, s_k),\ \forall s_k \in S,\ k = 0, 1, ..., N-1.$$

By Proposition 5.1, we have $p^{\hat{\pi}}_{s_0}(W) = \tilde{p}^{\hat{\pi}}_{(1,s_0)}(\{1\} \times W)$, $\forall s_0 \in S$. This implies that for every initial condition $s_0 \in S$,

$$\begin{aligned}
p^{\hat{\pi}}_{s_0}(W) =& \int_{\tilde{S}^{N+1}} \mathbf{1}_{\{1\}\times W}(\tilde{s}_N) \prod_{k=0}^{N-1} \tilde{\nu}(d\tilde{s}_{k+1}|\tilde{s}_k, \hat{\pi}_k(\tilde{s}_k)) \delta_{(1,s_0)}(d\tilde{s}_0) \\
=& \int_{\tilde{S}^N \times S} \mathbf{1}_W(s_N) \nu(ds_N|s_{N-1}, \hat{\pi}_{N-1}(1, s_{N-1})) \\
& \mathbf{1}_{\{1\}\times W}(\tilde{s}_{N-1}) \prod_{k=0}^{N-2} \tilde{\nu}(d\tilde{s}_{k+1}|\tilde{s}_k, \hat{\pi}_k(\tilde{s}_k)) \delta_{(1,s_0)}(d\tilde{s}_0) \\
=& \int_{\tilde{S}\times S^N} \left(\prod_{k=0}^{N-1} \mathbf{1}_W(s_{k+1}) \nu(ds_{k+1}|s_k, \hat{\pi}_k(1, s_k)) \right) \mathbf{1}_{\{1\}\times W}(\tilde{s}_0) \delta_{(1,s_0)}(d\tilde{s}_0) \\
=& \mathbf{1}_W(s_0) \int_{S^N} \prod_{k=0}^{N-1} \mathbf{1}_W(s_{k+1}) \nu(ds_{k+1}|s_k, \mu_k(s_k)) = p^{\mu}_{s_0}(W).
\end{aligned}$$

In other words, for every $\hat{\pi} \in \hat{\Pi}$, there exists a choice of policy $\mu \in \mathscr{M}$ that does at least as well. Thus, $p^{\hat{\pi}}_{s_0}(W) \leq \sup_{\mu\in\mathscr{M}} p^{\mu}_{s_0}(W)$, $\forall s_0 \in S, \hat{\pi} \in \hat{\Pi}$. The desired result then follows. □

We note briefly that in a perfect state information model, the sequence of functions $\bar{\eta}_k(\xi(p_0); i_k) = z_k$, $k \geq 0$ is not a sufficient statistic in the strict sense of Definition 5.4. However, due to the particular form of the cost function in the probabilistic safety problem, only decisions made with respect to safe trajectories (i.e. $h_k = 1$, $\forall k$) contribute to the final payoff. Thus, the controls for $h_k = 0$ can be chosen as identical to those for $h_k = 1$.

5.7 Specialization to Partially Observable Markov Decision Processes

In order to illustrate the state estimation and dynamic programming procedures for discrete models, we now consider a special case of the PObdtSHS model in which the state space, control input space, and observation space are finite. Namely, $S = Q$, $C_a = \Sigma$, and $Z = O$, for some finite sets Q, Σ, and O. This is commonly referred to in literature as Partially Observable Markov Decision Processes (see for example Russell and Norvig, 2002; Thrun et al., 2005).

Given a finite state space, the state transition kernel ν can be summarized in terms of a transition probability $p_q : Q \times \Sigma \times Q \to [0,1]$ such that

$$\nu(Q'|q,\sigma) = \sum_{q'\in Q'} p_q(q'|q,\sigma), \ \forall Q' \subseteq Q, q \in Q, \sigma \in \Sigma.$$

For simplicity of notation, we assume that the observations are not affected by the control inputs. In this case, the observation kernels can be summarized in terms of an observation probability

$p_o : Q \times O \to [0,1]$ such that

$$\zeta_0(O'|q) = \zeta(O'|q,\sigma) = \sum_{o' \in O'} p_o(o'|q),\ \forall O' \subseteq O, q \in Q, \sigma \in \Sigma.$$

We denote the POMDP model specified above as $\mathscr{H}_{\text{POMDP}} = (Q, \Sigma, O, p_q, p_o)$. For a given safe set $W \subseteq Q$, the corresponding augmented system model is given by $\tilde{\mathscr{H}}_{\text{POMDP}} = (\tilde{Q}, \Sigma, O, \tilde{p}_q, \tilde{p}_o)$, where $\tilde{Q}$, $\tilde{p}_q$ and $\tilde{p}_o$ are defined as

$$\tilde{Q} = \{0,1\} \times Q \tag{5.34}$$

$$\tilde{p}_q(h_{k+1}, q_{k+1}|(h_k, q_k), \sigma_k) = \begin{cases} p_q(q_{k+1}|q_k, \sigma_k), & h_k = 0, h_{k+1} = 0 \\ 0, & h_k = 0, h_{k+1} = 1 \\ \mathbf{1}_{Q \setminus W}(q_k) p_q(q_{k+1}|q_k, \sigma_k), & h_k = 1, h_{k+1} = 0 \\ \mathbf{1}_W(q_k) p_q(q_{k+1}|q_k, \sigma_k), & h_k = 1, h_{k+1} = 1, \end{cases}$$

$$\tilde{p}_o(o_k|h_k, q_k) = p_o(o_k|q_k),\ h_k = 0, 1.$$

for every $q_k, q_{k+1} \in Q$ and $o_k \in O$. As discussed in section 5.3.1, for a given initial state distribution $p_0 \in \mathscr{P}(Q)$ and policy $\tilde{\pi}' \in \tilde{\Pi}'$, the transition probability $\tilde{p}_q$ and observation probability $\tilde{p}_o$ induce a unique probability measure $\tilde{P}_k(\tilde{\pi}', \xi(p_0))$ on the sample space $\tilde{\Omega}_k := \tilde{Q}^{k+1} \times O^{k+1} \times \Sigma^k$.

First we will show that the state filtering equations in (5.14) in this case simplifies to a Bayesian update rule for the augmented POMDP $\tilde{\mathscr{H}}_{\text{POMDP}}$. The precise statement is as follows.

Lemma 5.1. *Let $\mathscr{H}_{POMDP}$ be a POMDP and $W \subseteq Q$ be a safe set. Let $\tilde{\mathscr{H}}_{POMDP}$ be the corresponding augmented POMDP. Then for every $p_0 \in \mathscr{P}(Q)$, $\tilde{\pi}' \in \tilde{\Pi}'$, $\tilde{q}_k \in \tilde{Q}$, and $k = 0, 1, ...,$ we have*

$$\tilde{p}_0(\xi(p_0); i_0)(\tilde{q}_0) = \frac{\tilde{p}_o(o_0|\tilde{q}_0)\xi(p_0)(\tilde{q}_0)}{\sum_{\tilde{q}_0 \in \tilde{Q}} \tilde{p}_o(o_0|\tilde{q}_0)\xi(p_0)(\tilde{q}_0)},$$

$$\tilde{p}_k(\xi(p_0); i_k)(\tilde{q}_k) = \frac{\tilde{p}_o(o_k|\tilde{q}_k)\tilde{p}_{k|k-1}(\xi(p_0); i_{k-1}, \sigma_{k-1})(\tilde{q}_k)}{\sum_{\tilde{q}_k \in \tilde{Q}} \tilde{p}_o(o_k|\tilde{q}_k)\tilde{p}_{k|k-1}(\xi(p_0); i_{k-1}, \sigma_{k-1})(\tilde{q}_k)},\ k \geq 1,$$

where

$$\tilde{p}_{k|k-1}(\xi(p_0); i_{k-1}, \sigma_{k-1})(\tilde{q}_k) = \sum_{\tilde{q}_{k-1} \in \tilde{Q}} \tilde{p}_q(\tilde{q}_k|\tilde{q}_{k-1}, \sigma_{k-1})\tilde{p}_{k-1}(\xi(p_0); i_{k-1})(\tilde{q}_{k-1}),$$

for $\tilde{P}_k(\tilde{\pi}', \xi(p_0))$ almost every i_k.

The proof of this result largely revolves around manipulating the abstract definitions of section 5.3.2, and can be found in appendix C. Using the transition probability $\tilde{p}_q$ and observation probability $\tilde{p}_o$, the filtering equations in the statement of Lemma 5.1 can be rewritten as follows.

1. Initialization Step:

$$\tilde{p}_0(\xi(p_0);i_0)(1,q_0) = \frac{p_o(o_0|q_0)p(o_0)}{\sum_{q_0\in Q} p_o(o_0|q_0)p(q_0)}, \tag{5.35}$$
$$\tilde{p}_0(\xi(p_0);i_0)(0,q_0) = 0;$$

2. Prediction Step:

$$\tilde{p}_{k+1|k}(\xi(p_0);i_k,\sigma_k)(1,q_{k+1}) = \sum_{q_k\in W} p_q(q_{k+1}|q_k,\sigma_k)\tilde{p}_k(\xi(p_0);i_k)(1,q_k), \tag{5.36}$$

$$\begin{aligned}\tilde{p}_{k+1|k}(\xi(p_0);i_k,\sigma_k)(0,q_{k+1}) = &\sum_{q_k\in Q} p_q(q_{k+1}|q_k,\sigma_k)\tilde{p}_k(\xi(p_0);i_k)(0,q_k) \\ &+ \sum_{q_k\in Q\setminus W} p_q(q_{k+1}|q_k,\sigma_k)\tilde{p}_k(\xi(p_0);i_k)(1,q_k)\end{aligned} \tag{5.37}$$

3. Update Step:

$$\begin{aligned}&\tilde{p}_{k+1}(\xi(p_0);i_{k+1})(h_{k+1},q_{k+1}) = \\ &\frac{p_o(o_{k+1}|q_{k+1})\tilde{p}_{k+1|k}(\xi(p_0);i_k,\sigma_k)(h_{k+1},q_{k+1})}{\sum_{(h_{k+1},q_{k+1})\in\tilde{Q}} p_o(o_{k+1}|q_{k+1})\tilde{p}_{k+1|k}(\xi(p_0);i_k,\sigma_k)(h_{k+1},q_{k+1})}.\end{aligned} \tag{5.38}$$

From the Bayesian update equations (5.35)–(5.38), it can be observed that estimation in the case of a partial information safety problem for the POMDP involves maintaining a discrete probability distribution over an augmented state space $\tilde{Q}$ containing twice the number of discrete states as the original state space Q. In particular, if one were to marginalize the augmented distribution $\tilde{p}_k$ over the history state h_k, one recovers the regular POMDP update equations for the conditional distribution of q_k. More specifically, let $p_{k|k}(\cdot|p_0;i_k)$ be the conditional state distribution of q_k over Q, given the initial distribution p_0 and the information vector i_k. The update equations for $p_{k|k}$ can be found in for example Chapter 17 of Russell and Norvig (2002) or Chapter 15 of Thrun et al. (2005). Then it can be verified that

$$p_{k|k}(q_k|p_0;i_k) = \tilde{p}_k(\xi(p_0);i_k)(1,q_k) + \tilde{p}_k(\xi(p_0);i_k)(0,q_k),\ \forall q_k \in Q.$$

Thus, the augmented information state $\tilde{p}_k$ provides slightly more information about the history of state evolution (i.e. the safety of past trajectory) as compared with the regular information state $p_{k|k}$. As we will illustrate through an example later on, this extra information can be important in a probabilistic safety problem.

In order to give a compact statement of the dynamic programming equations, consider a Borel-measurable mapping $f : \mathscr{P}(\tilde{Q}) \times O \times \Sigma \to \mathscr{P}(\tilde{Q})$ defined as

$$f(\tilde{p},o,\sigma)(h',q') = \Phi(h',q'|\Psi(\tilde{p},\sigma);o,\sigma),\ (h',q') \in \tilde{Q},$$

where Φ and Ψ are the abstract filtering operators in (5.12) and (5.13), respectively. Then we have that $\tilde{p}_k$ is recursively updated according to

$$\tilde{p}_{k+1}(\xi(p_0);i_{k+1}) = f(\tilde{p}_k(\xi(p_0);i_k), o_{k+1}, \sigma_k),$$

initialized with $\tilde{p}_0(\xi(p_0);i_0) = \Phi_0(\cdot|\xi(p_0);i_0)$, where Φ_0 is as defined in (5.11). By the result of Lemma 5.1, on a set of executions which occur with probability one, $\tilde{p}_0(\xi(p);i_0)$ has the representation in equation (5.35) and the operator f has the representation in equations (5.36), (5.37), and (5.38).

Specializing (5.15), the transition kernel $\hat{\nu}$ characterizing the evolution of the information state is given by

$$\hat{\nu}(E|\tilde{p},\sigma) = \sum_{\tilde{q}\in\tilde{Q}} \sum_{\tilde{q}'\in\tilde{Q}} \sum_{o\in G(\tilde{p},\sigma,E)} \tilde{p}_o(o|\tilde{q}')\tilde{p}_q(\tilde{q}'|\tilde{q},\sigma)\tilde{p}(\tilde{q}),$$

where $G(\tilde{p},\sigma,E) = \{o \in O : f(\tilde{p},o,\sigma) \in E\}$, for $\tilde{p} \in \hat{Q}$, $\sigma \in \Sigma$, and $E \in \mathscr{B}(\hat{Q})$. In practical terms, the dynamic programming operator $\mathscr{T}_{Safe}$ defined in (5.20) can be rewritten for POMDPs as

$$\mathscr{T}_{Safe}(J)(\tilde{p}) = \max_{\sigma\in\Sigma} \sum_{(h,q)\in\tilde{Q}} \sum_{q'\in Q} \sum_{o\in O} J(f(\tilde{p},o,\sigma))p_o(o|q')p_q(q'|q,\sigma)\tilde{p}(h,q),\ \tilde{p} \in \hat{Q}. \tag{5.39}$$

By Theorem 5.1, the maximal probability of safety over a finite time horizon $[0,N]$ is computed by the following dynamic programming recursion:

$$p^*(p_0;W) = \sum_{q_0\in Q} \sum_{o_0\in O} \mathscr{T}^N_{Safe}(g_N)(\tilde{p}_0(\xi(p_0);o_0))p_o(o_0|q_0)p_0(q_0) \tag{5.40}$$

with the terminal cost

$$g_N(\tilde{p}_N) = \sum_{q_N\in W} \tilde{p}_N(1,q_N). \tag{5.41}$$

By the form of the dynamic programming equations (5.39)–(5.41), the partial information safety problem is a terminal cost problem for the augmented POMDP model $\tilde{\mathscr{H}}_{\text{POMDP}}$ with twice the number of discrete states as the original POMDP model $\mathscr{H}_{\text{POMDP}}$. Thus, in principle, one can apply existing computational techniques for POMDP problems to find the optimal control policy (Russell and Norvig, 2002, Chapter 17; Thrun et al., 2005, Chapter 15). However, as shown by Papadimitriou and Tsitsiklis (1987) and Lusena et al. (2001), the problem of computing optimal or even ε-optimal control policies for POMDPs is in general PSPACE-complete (note that NP is a subset of PSPACE), due to the possible exponential increase in complexity of the optimal policy with respect to the number of discrete states, observations, control actions, and time steps. As such, exact optimal policies are typically computed only for models with no more than about 20 discrete states.

In the following, we will illustrate these procedures using a concrete example. Consider a POMDP $\mathscr{H}_{\text{POMDP}}$ with state space $Q = \{q_1, q_2, q_3, q_4\}$, control input space $\Sigma = \{\sigma_L, \sigma_R\}$, and observation space $O = \{o_L, o_R\}$. The transition probability function p_q for $\mathscr{H}_{\text{POMDP}}$ can be described

in terms of a transition probability matrix $P_q(\sigma) \in [0,1]^{4\times 4}$ such that $p_q(q_j|q_i,\sigma) = P_q(\sigma)(i,j)$ for every $q_i, q_j \in Q$ and $\sigma \in \Sigma$. For this example, $P_q(\sigma)$ is given below, with the corresponding state transition diagram as shown in Figure 5.3.

$$P_q(\sigma_L) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, P_q(\sigma_R) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The corresponding state transition diagram is given by

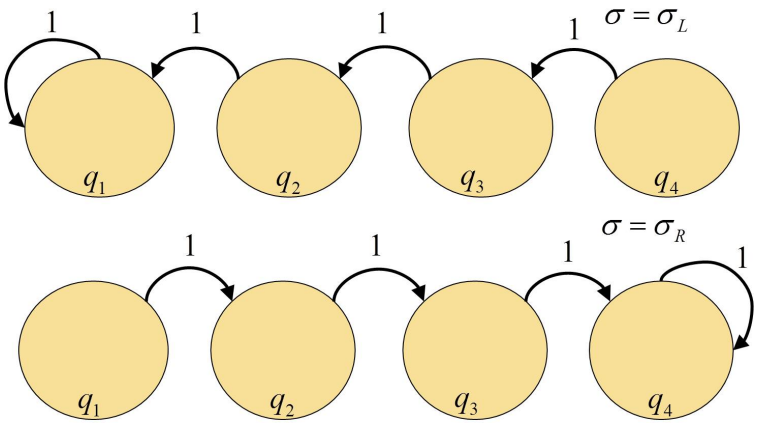


Figure 5.3: State transition diagram for POMDP example.

The observation probability function p_o for $\mathscr{H}_{\text{POMDP}}$ can be described in terms of an observation probability matrix $P_o(o) \in [0,1]^{4\times 4}$ such that $p_o(o|q_i) = P_o(o)(i,i)$ for every $q_i \in Q$ and $o \in O$. For this example, $P_o(o)$ is specified as follows.

$$P_o(o_L) = \text{diag}\left(\ \alpha,\quad \alpha,\quad 1-\alpha,\quad 1-\alpha\ \right),$$
$$P_o(o_R) = \text{diag}\left(\ 1-\alpha,\quad 1-\alpha,\quad \alpha,\quad \alpha\ \right).$$

where $\alpha \in [0,1]$. The corresponding state observation diagram is as shown in Figure 5.4.

We consider an initial state distribution p_0 described in terms of a vector $\bar{p}_0 \in [0,1]^4$ such that $p_0(q_i) = \bar{p}_0(i), \forall q_i \in Q$:

$$\bar{p}_0 = \left[\ \tfrac{1-\beta}{2} \quad \tfrac{\beta}{2} \quad \tfrac{\beta}{2} \quad \tfrac{1-\beta}{2}\ \right]^T,$$

where $\beta \in [0,1]$. The safe set is selected to be $W = \{q_2, q_3\}$.

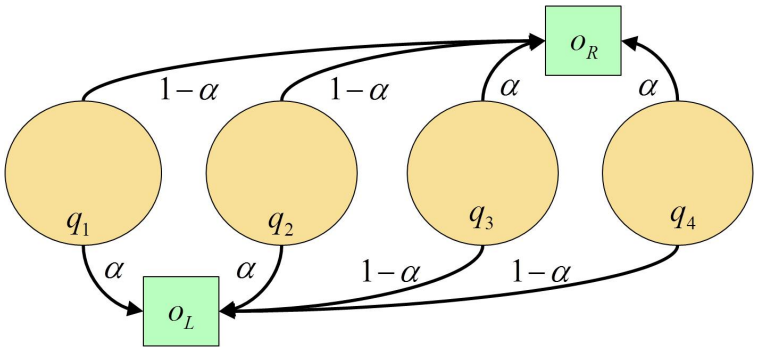


Figure 5.4: State observation diagram for POMDP example.

With a representation of $\tilde{p}_k(\xi(p_0);i_0)(h,\cdot)$ by vectors $\tilde{p}_k^h \in [0,1]^4$, $h=0,1$, the state filtering equations given in (5.35)–(5.38) can be rewritten in a compact form:

$$\tilde{p}_0^1 = \frac{P_o(o_0)\bar{p}_0}{\mathbf{1}^T P_o(o_0)\bar{p}_0},\ \tilde{p}_0^0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T;$$
$$\tilde{p}_{k+1}^1 = \frac{P_o(o_{k+1})P_q(\sigma_k)^T M_W \tilde{p}_k^1}{\mathbf{1}^T P_o(o_{k+1})P_q(\sigma_k)^T(\tilde{p}_k^0+\tilde{p}_k^1)},$$
$$\tilde{p}_{k+1}^0 = \frac{P_o(o_{k+1})P_q(\sigma_k)^T(\tilde{p}_k^0+M_{Q\setminus W}\tilde{p}_k^1)}{\mathbf{1}^T P_o(o_{k+1})P_q(\sigma_k)^T(\tilde{p}_k^0+\tilde{p}_k^1)},$$

where $\mathbf{1} = [1\ 1\ 1\ 1]^T$ and $M_W,\ M_{Q\setminus W} \in [0,1]^{4\times 4}$ are diagonal matrices given by

$$M_W = \mathrm{diag}(0,\ 1,\ 1,\ 0),\ M_{Q\setminus W} = \mathrm{diag}(1,\ 0,\ 0,\ 1).$$

Given a time horizon $[0,N]$, $N \geq 1$, the first step of the dynamic programming procedure described in (5.39) can be carried out as follows.

$$J^*_{N-1\to N}(\tilde{p}) = \mathscr{T}_{Safe}(g_N)(\tilde{p}) = \max_{\sigma\in\Sigma}\sum_{o\in O} g_N(f(\tilde{p},o,\sigma))\mathbf{1}^T P_o(o)P_q(\sigma)^T(\tilde{p}^0+\tilde{p}^1),$$

where $\tilde{p}^h$ denotes the vector $\tilde{p}(h,\cdot)\in[0,1]^4$. By the definitions of g_N and f, we then have

$$J^*_{N-1\to N}(\tilde{p}) = \max_{\sigma\in\Sigma}\sum_{o\in O}\sum_{q_j\in W}(P_o(o)P_q(\sigma)^T M_W\tilde{p}^1)(j) = \max_{\sigma\in\Sigma}\sum_{q_j\in W}(P_q(\sigma)^T M_W\tilde{p}^1)(j),$$

where we use the fact that $\sum_o P_o(o) = \mathbf{I}$. It then follows by the definitions of $P_q(\sigma)$ and M_W that

$$J^*_{N-1\to N}(\tilde{p}) = \max_{\sigma\in\Sigma}\sum_{q_i\in W}\sum_{q_j\in W} p_q(q_j|q_i,\sigma)\tilde{p}(1,q_i) = \begin{cases}\tilde{p}(1,q_2), & \tilde{p}(1,q_2)\geq\tilde{p}(1,q_3)\\ \tilde{p}(1,q_3), & \text{otherwise,}\end{cases}$$

with the corresponding optimal control policy

$$\hat{\pi}^*_{N-1}(\tilde{p}) = \begin{cases} \sigma_R, & \tilde{p}(1,q_2) \geq \tilde{p}(1,q_3) \\ \sigma_L, & \text{otherwise.} \end{cases}$$

Using a similar line of reasoning, we can show that the optimal cost-to-go function at time $k = N-2$ is given by

$$J^*_{N-2\to N} = \mathscr{T}_{Safe}(J^*_{N-1\to N}) = J^*_{N-1\to N}.$$

Thus, $J^*_{N-1\to N}$ is a fixed point of the operator $\mathscr{T}_{Safe}$ and

$$J^*_N(\tilde{p}) = J^*_{0\to N}(\tilde{p}) = J^*_{N-1\to N}(\tilde{p}),\ \forall \tilde{p} \in \hat{Q},\ N \geq 1.$$

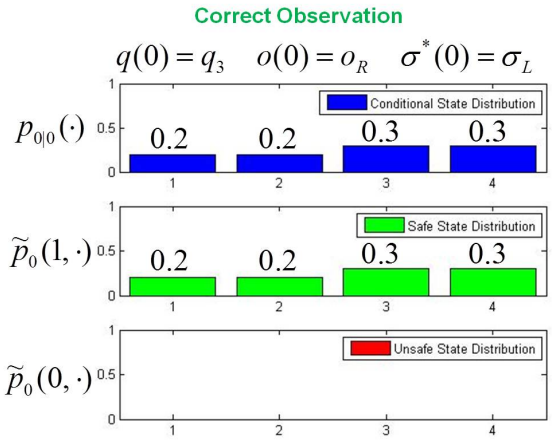
Furthermore, a stationary optimal control policy is given by $\hat{\pi}^* = (\hat{\pi}^*_{N-1}, \hat{\pi}^*_{N-1}, ..., \hat{\pi}^*_{N-1})$. The corresponding maximal probability of safety is computed as

$$p^*(p_0; W) = \sum_{o_0 \in O} J^*_N(\tilde{p}_0(\xi(p_0); o_0))\mathbf{1}^T P_o(o_0)\tilde{p}_0 = \begin{cases} \alpha\beta, & \alpha \geq 0.5 \\ (1-\alpha)\beta, & \text{otherwise.} \end{cases}$$

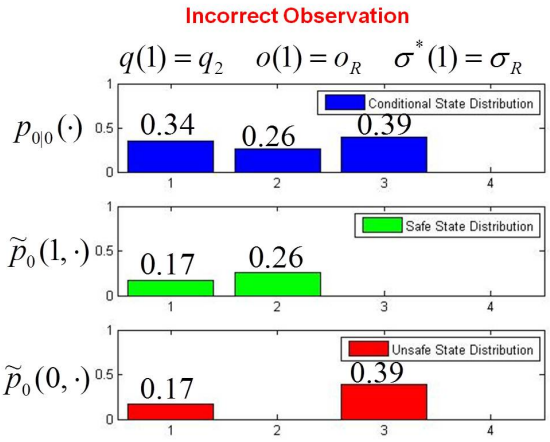
To demonstrate the value of the augmented information state, we plot in Figure 5.5 the realizations of $p_{k|k}$ for the original model $\mathscr{H}_{\text{POMDP}}$ and $\tilde{p}_k$ for the augmented model $\tilde{\mathscr{H}}_{\text{POMDP}}$ in a sample simulation run under the optimal policy $\hat{\pi}^*$ over time steps $k = 0,1,2$. In this simulation, the parameters of the model were chosen to be $\alpha = 0.6$ and $\beta = 0.5$. At the initial step, the information provided by the regular information state $p_{0|0}$ and the augmented information state $\tilde{p}_0$ are essentially the same, namely $p_{0|0}$ is the component of $\tilde{p}_0$ corresponding to $h_0 = 1$. However, at time step $k = 1$, due to an erroneous measurement, the regular information state $p_{0|0}$ would seem to suggest that the most likely system state is q_3, while the actual system state q_2 has the least likelihood out of all the states with non-zero probability. If one were to select an input according to this belief, then perhaps one would choose $\sigma(1) = \sigma_L$, which would render the actual state trajectory unsafe. On the other hand, the augmented information state splits the distribution $p_{0|0}$ in two, and weights the conditional probability in each component according to the likelihood that the trajectory has been safe or unsafe. In particular, the safe component ($h_1 = 1$) of $\tilde{p}_1$ takes into account the fact that given the last input was $\sigma(0) = \sigma_L$, the state trajectory could not have been safe if the current system state were q_3. Thus, the state q_3 is given a zero weighting in the safe component, resulting in a correct choice of input $\sigma^*(1) = \sigma_R$ according to $\hat{\pi}^*$. In fact, one can show that, for this particular example, as long as the system state is initialized in a safe state (either q_2 or q_3), and a correct observation is obtained at time $k = 0$, then the optimal policy $\hat{\pi}^*$ would ensure a correct choice of input for all $k \geq 0$, regardless of the realization of the output trajectory.

5.8 Specialization to Probability Density Models of Stochastic Hybrid Systems

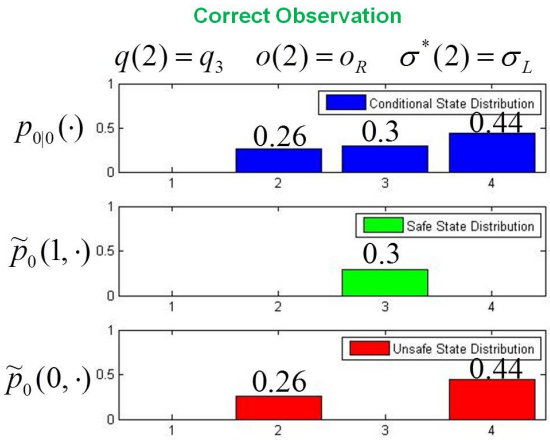
In this section, we consider the more general case of a POdtSHS equipped with a hybrid state space $S := \bigcup_{q\in Q}\{q\} \times \mathbb{R}^{n(q)}$ and a hybrid observation space $Z = O \times \bigcup_{q\in Q}\mathbb{R}^{n_o(q)}$, where $O :=$



(a) Information states at $k = 0$



(b) Information states at $k = 1$



(c) Information states at $k = 2$

Figure 5.5: Sample simulation run of POMDP example over three time steps.

$\{o_1, o_2, ..., o_{m'}\}$ is the discrete observation space and $n_o : Q \to \mathbb{N}$ is the dimension of the continuous observation space. To avoid technical complications, it is assumed that the evolution of the continuous state and the generation of continuous observations in this system are modeled by non-degenerate probability distributions, so that the stochastic kernels ν_x, ν_r, ζ_0, and ζ can be described in terms of probability density functions on Euclidean spaces. More precisely, the assumption on $\mathscr{H}$ is as follows.

Assumption 5.1.

1. There exists a Borel-measurable probability density function $p_x : \mathbb{R}^{n(\cdot)} \times S \times C_a \to \mathbb{R}$ such that for each $s = (q,x) \in S$ and $a \in C_a$,

$$\nu_x(X'|s,a) = \int_{X'} p_x(x'|(q,x),a)dx', \ \forall X' \in \mathscr{B}(\mathbb{R}^{n(q)});$$

2. There exists a Borel-measurable probability density function $p_r : \mathbb{R}^{n(\cdot)} \times S \times C_a \times Q \to \mathbb{R}$ such that for each $s = (q,x) \in S$, $a \in C_a$, and $q' \in Q$,

$$\nu_r(X'|s,a,q') = \int_{X'} p_r(x'|(q,x),a,q')dx', \ \forall X' \in \mathscr{B}(\mathbb{R}^{n(q')});$$

3. There exists a Borel-measurable probability density function $p_{z,0} : Z \times S \to \mathbb{R}$ such that for each $s = (q,x) \in S$,

$$\zeta_0(\{o\} \times Y'|s) = \int_{Y'} p_{z,0}((o,y')|(q,x))dy', \ \forall o \in O, \ Y' \in \mathscr{B}(\mathbb{R}^{n_o(q)});$$

4. There exists a Borel-measurable probability density function $p_z : Z \times S \times C_a \to \mathbb{R}$ such that for each $s = (q,x) \in S$ and $a \in C_a$,

$$\zeta(\{o\} \times Y'|s,a) = \int_{Y'} p_z((o,y')|(q,x),a)dy', \ \forall o \in O, \ Y' \in \mathscr{B}(\mathbb{R}^{n_o(q)}).$$

Using the density functions p_x and p_r, we can define, for each $s = (q,x) \in S$ and $a \in C_a$, a Borel-measurable hybrid probability density function $p_s(s'|s,a)$ as follows

$$p_s((q',x')|(q,x),a) = \begin{cases} \nu_q(q|(q,x),a)p_x(x'|(q,x),a), & \text{if } q' = q \\ \nu_q(q'|(q,x),a)p_r(x'|(q,x),a,q'), & \text{if } q' \neq q. \end{cases}$$

Let $S' = \bigcup_{q \in Q} \{q\} \times X_q$ be any Borel subset of the hybrid state space S such that $X_q \in \mathscr{B}(\mathbb{R}^{n(q)})$, $\forall q \in Q$. Then the hybrid state transition kernel ν can be characterized in terms of the probability density function p_s as

$$\nu(S'|s,a) = \sum_{q' \in Q} \int_{X_{q'}} p_s((q',x')|(q,x),a)dx', \ \forall s = (q,x) \in S, \ a \in C_a.$$

Thus, under Assumption 5.1, a POdtSHS with hybrid observation space can be equivalently characterized in terms of the tuple $\mathscr{H} = (S, C_a, Z, p_s, p_{z,0}, p_z)$. We will refer to this as a probability density model of POdtSHS. It can be observed that the probability density functions p_s, $p_{z,0}$, and p_z are analogous to the probability mass functions p_q and p_o in the POMDP case. For a given safe set $W \in \mathscr{B}(S)$, the corresponding augmented system model $\tilde{\mathscr{H}} = (\tilde{S}, C_a, Z, \tilde{p}_s, \tilde{p}_{z,0}, \tilde{p}_z)$ can be defined similarly as in (5.34).

To be consistent with Assumption 5.1, we will consider initial state distributions $p_0 \in \mathscr{P}(S)$ which can be characterized in terms of a Borel-measurable probability density function $\bar{p}_0 : S \to \mathbb{R}$ such that

$$p_0(\{q\} \times X') = \int_{X'} \bar{p}_0(q,x)dx, \ \forall q \in Q, \ X' \in \mathscr{B}(\mathbb{R}^{n(q)}).$$

With a slight abuse of notation, we will denote by $\xi(\bar{p}_0)$ the probability density function associated with the initial state distribution $\xi(p_0)$ on the augmented state space. Then, given a policy $\tilde{\pi}' \in \tilde{\Pi}'$, the probability density functions $\bar{p}_0$, $\tilde{p}_s$, $\tilde{p}_{z,0}$, and $\tilde{p}_z$ induce a unique probability measure $\tilde{P}_k(\tilde{\pi}', \xi(\bar{p}_0))$ on the sample space $\tilde{\Omega}_k := \tilde{S}^{k+1} \times Z^{k+1} \times C_a^k$.

It turns out that, for the class of POdtSHS considered here, the filtering equations given in section 5.3.2 specializes to the Bayesian update rule for the conditional probability density of the augmented state $\tilde{s}_k$ given $i_k \in I_k$ and $p_0 \in \mathscr{P}(\tilde{S})$. This is analogous to the update rule for the conditional probability mass function in the POMDP case. For compactness of notation, we denote the integration of a Borel-measurable function $\tilde{F} : \tilde{S} \to \mathbb{R}$ over $\tilde{S}$ as

$$\int_{\tilde{S}} \tilde{F}(\tilde{s})d\tilde{s} := \sum_{h\in\{0,1\}} \sum_{q\in Q} \int_{\mathbb{R}^{n(q)}} \tilde{F}(h,q,x)dx.$$

Lemma 5.2. *Let $\mathscr{H} = (S, C_a, Z, p_s, p_{z,0}, p_z)$ be a probability density model of POdtSHS and $W \in \mathscr{B}(S)$ be a Borel safe set. Let $\tilde{\mathscr{H}} = (\tilde{S}, C_a, Z, \tilde{p}_s, \tilde{p}_{z,0}, \tilde{p}_z))$ be the corresponding augmented POdtSHS. Let $\bar{p}_0 : S \to \mathbb{R}$ be a Borel-measurable initial state density. Then for every $\tilde{\pi}' \in \tilde{\Pi}'$ and $k = 0, 1, ...,$ the stochastic kernel $\tilde{p}_k(\xi(\bar{p}_0); i_k)$ has the probability density $\tilde{p}_k^d(\cdot|\xi(\bar{p}_0); i_k) : \tilde{S} \to \mathbb{R}$ given by*

$$\tilde{p}_0^d(\tilde{s}_0|\xi(\bar{p}_0); i_0) = \frac{\tilde{p}_{z,0}(z_0|\tilde{s}_0)\xi(\bar{p}_0)(\tilde{s}_0)}{\int_{\tilde{S}} \tilde{p}_{z,0}(z_0|\tilde{s}_0')\xi(\bar{p}_0)(\tilde{s}_0')d\tilde{s}_0'},$$

$$\tilde{p}_k^d(\tilde{s}_k|\xi(\bar{p}_0); i_k) = \frac{\tilde{p}_z(z_k|\tilde{s}_k, a_{k-1})\tilde{p}_{k|k-1}^d(\tilde{s}_k|\xi(\bar{p}_0); i_{k-1}, a_{k-1})}{\int_{\tilde{S}} \tilde{p}_z(z_k|\tilde{s}_k', a_{k-1})\tilde{p}_{k|k-1}(\tilde{s}_k'|\xi(\bar{p}_0); i_{k-1}, a_{k-1})d\tilde{s}_k'},$$

where

$$\tilde{p}_{k|k-1}^d(\tilde{s}_k|\xi(\bar{p}_0); i_{k-1}, a_{k-1}) = \int_{\tilde{S}} \tilde{p}_s(\tilde{s}_k|\tilde{s}_{k-1}, a_{k-1})\tilde{p}_{k-1}^d(\tilde{s}_{k-1}|\xi(\bar{p}_0); i_{k-1})d\tilde{s}_{k-1},$$

for $\tilde{P}_k(\tilde{\pi}', \xi(\bar{p}_0))$ almost every i_k.

The proof of this result is again somewhat technical in nature and can be found in appendix D. Using the definitions of $\tilde{p}_s$, $\tilde{p}_{z,0}$, and $\tilde{p}_z$, the update equations for the probability density in the statement of Lemma 5.2 can be rewritten as follows.

1. Initialization Step:

$$\tilde{p}_0^d(1,s_0|\xi(\bar{p}_0);i_0) = \frac{p_{z,0}(z_0|s_0)\bar{p}_0(s_0)}{\int_S p_{z,0}(z_0|s_0)\bar{p}(s_0)ds_0}, \tag{5.42}$$
$$\tilde{p}_0^d(0,s_0|\xi(\bar{p}_0);i_0) = 0;$$

2. Prediction Step:

$$\tilde{p}_{k+1|k}^d(\xi(\bar{p}_0);i_k,a_k)(1,s_{k+1}) = \int_W p_s(s_{k+1}|s_k,a_k)\tilde{p}_0^d(1,s_k|\xi(\bar{p}_0);i_k)ds_k, \tag{5.43}$$
$$\begin{aligned}\tilde{p}_{k+1|k}^d(\xi(\bar{p}_0);i_k,a_k)(0,s_{k+1}) = &\int_S p_s(s_{k+1}|s_k,a_k)\tilde{p}_0^d(0,s_k|\xi(\bar{p}_0);i_k)ds_k \\ &+ \int_{S\setminus W} p_s(s_{k+1}|s_k,a_k)\tilde{p}_0^d(1,s_k|\xi(\bar{p}_k);i_k)ds_k\end{aligned} \tag{5.44}$$

3. Update Step:

$$\begin{aligned}&\tilde{p}_{k+1}^d(h_{k+1},s_{k+1}|\xi(\bar{p}_0);i_{k+1}) = \\ &\frac{p_z(z_{k+1}|s_{k+1},a_k)\tilde{p}_{k+1|k}^d(\xi(\bar{p}_0);i_k,a_k)(h_{k+1},s_{k+1})}{\sum_{h_{k+1}\in\{0,1\}}\int_S p_z(z_{k+1}|s_{k+1},a_k)\tilde{p}_{k+1|k}^d(\xi(\bar{p}_0);i_k,a_k)(h_{k+1},s_{k+1})ds_{k+1}}.\end{aligned} \tag{5.45}$$

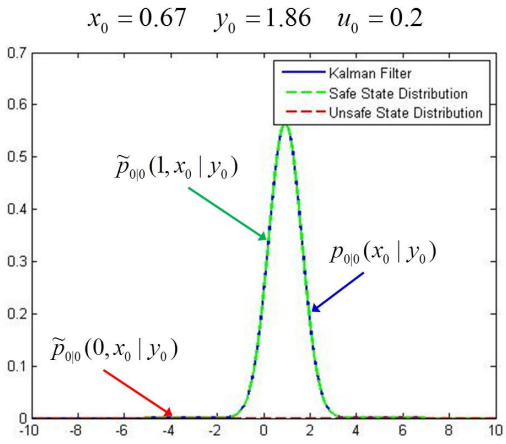
Similarly as in the case of a POMDP, the estimation procedure given in (5.42)–(5.45) for a probability density model of POdtSHS involves maintaining a conditional probability density function over a hybrid state space with twice the number of discrete states as in the original hybrid system model. However, while the probability distribution for a POMDP is a vector of probabilities, and hence finite dimensional, the hybrid probability density is a real-valued function over the continuous state space within each mode, and hence infinite dimensional. It can be also verified that marginalizing over the history state h_k recovers the conditional probability density function of the hybrid state s_k, as produced by Bayesian update equations (see for example Kumar and Varaiya, 1986, chapter 5). Specifically, let $p_{k|k}^d(\cdot|p_0;i_k)$ be the conditional probability density of s_k over S, given the initial probability density $\bar{p}_0$ and the information vector i_k. Then we have

$$p_{k|k}^d(s_k|\bar{p}_0;i_k) = \tilde{p}_k^d(\xi(\bar{p}_0);i_k)(1,s_k) + \tilde{p}_k^d(\xi(\bar{p}_0);i_k)(0,s_k),\ \forall s_k \in S.$$

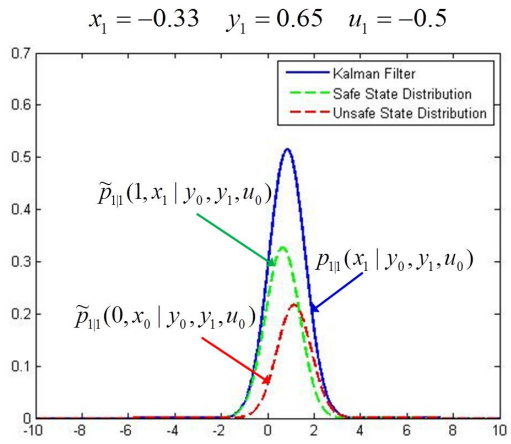
In essence, the augmented probability density $\tilde{p}_k$ splits the regular probability density into two components, and weights each according to the likelihood that the past trajectory has been safe. To illustrate the form of the augmented probability density, we consider a simple example of 1-D linear Gaussian system, as described below:

$$\begin{aligned}x_{k+1} &= x_k + u_k + w_k,\\ y_k &= x_k + v_k,\end{aligned}$$

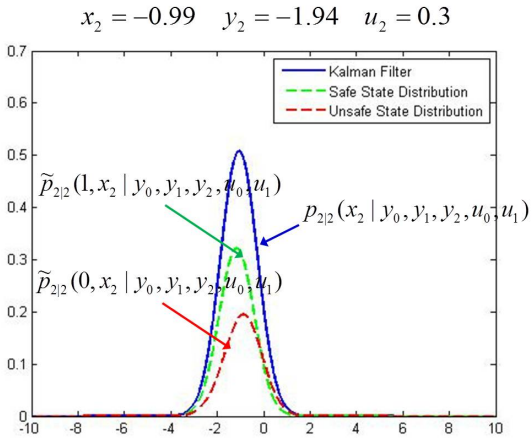
where x_0, w_k, v_k are random variables with standard normal distribution $\mathcal{N}(0,1)$. The probability density model of this system can be then derived as $p_x(x_{k+1}|x_k,u_k) = \mathcal{N}(x_k+u_k,1)$ and $p_z(y_k|x_k,u_{k-1}) = \mathcal{N}(x_k,1)$ (the reset kernel coincide with the continuous state transitional kernel given that there is only one mode). It can be seen that the regular information state $p_{k|k}$ in this case is simply the output of the Kalman filter for $\mathscr{H}$. Selecting the safe set as the interval $W = [-1,1]$, we simulate the system forward in time and plot the realizations of the Kalman filter output and the augmented information state $\tilde{p}_k$ for $\tilde{\mathscr{H}}$, under somewhat arbitrary choices of control input. In this case, the components of the augmented density function $\tilde{p}_k$ are computed by numerical integration using equations (5.42)–(5.45). The results of a sample simulation run over time steps $k = 0,1,2$ are shown in Figure 5.6.



(a) Information states at $k = 0$



(b) Information states at $k = 1$



(c) Information states at $k = 2$

Figure 5.6: Sample simulation run of linear Gaussian example over three time steps.

As in the case of discrete state systems, the information provided by the Kalman filter and the augmented probability density are the same at the first time step, namely the safe component ($h_0 = 1$) of the augmented probability density coincide with the regular conditional density. However, at the second or third time step, the augmented probability density splits the Kalman filter into two components according to the likelihood that the state history x_0 or (x_0, x_1) was in the safe set. Note that this is a hybrid probability density on the augmented state space $\{0,1\} \times \mathbb{R}$, even though the original system is continuous. While the output of a Kalman filter is Gaussian, it is no longer clear whether the components of the augmented density are in fact Gaussian, although they appear to exhibit Gaussian characteristics in Figure 5.6.

To derive a compact form of the dynamic programming equations, we again consider a Borel-measurable mapping $f : \mathscr{P}(\tilde{S}) \times Z \times C_a \to \mathscr{P}(\tilde{S})$ given by

$$f(\tilde{p}, z, a) = \Phi(d\tilde{s}|\Psi(\tilde{p}, a); z, a),$$

where Φ and Ψ are the abstract filtering operators in (5.12) and (5.13), respectively. Then we have that

$$\tilde{p}_{k+1}(\xi(\bar{p}_0); i_{k+1}) = f(\tilde{p}_k(\xi(\bar{p}_0); i_k), z_{k+1}, a_k),$$

initialized with $\tilde{p}_0(\xi(\bar{p}_0); i_0) = \Phi_0(\cdot|\xi(\bar{p}_0); z_0)$, where Φ_0 is as defined in (5.11). By the result of Lemma 5.2, on a set of executions which occur with probability one, $\tilde{p}_0(\xi(\bar{p}_0); i_0)$ has the probability density given in equation (5.42) and $\tilde{p}_k(\xi(\bar{p}_0); i_k)$, $k \geq 1$ has the probability density given in equation (5.45).

The dynamic programming operator $\mathscr{T}_{Safe}$ can be rewritten for a probability density model $\mathscr{H}$ of POdtSHS as follows.

$$\mathscr{T}_{Safe}(J)(\tilde{p}) = \sup_{a \in C_a} \sum_h \int_S \int_S \left(\int_Z J(f(\tilde{p}, z, a)) p_z(z|s', a) dz \right) p_s(s'|s, a) ds' \tilde{p}^d(h, s) ds, \tag{5.46}$$

where $\tilde{p}^d$ is the probability density associated with the information state $\tilde{p} \in \hat{S}$. By Theorem 5.1, the maximal probability of safety for $\mathscr{H}$ over time horizon $[0, N]$ is then computed by the following dynamic programming recursion:

$$p^*(\bar{p}_0; W) = \int_S \int_Z \mathscr{T}^N_{Safe}(g_N)(\tilde{p}_0(\xi(\bar{p}_0); z_0)) p_{z,0}(z_0|s_0) \bar{p}_0(s_0) dz_0 ds_0 \tag{5.47}$$

with the terminal cost

$$g_N(\tilde{p}_N) = \int_W \tilde{p}^d_N(1, s_N) ds_N. \tag{5.48}$$

As can be seen from equations (5.46)–(5.48), the computation of optimal safety probability for $\mathscr{H}$ in general needs to be carried out on the space of augmented probability density functions. Thus, finding computationally tractable algorithms for the synthesis of optimal control policies hinges on the existence of finite dimensional representations of the hybrid probability density $\tilde{p}^d_k$ in particular instances of the POdtSHS model.

Chapter 6

Conclusions

In many application scenarios found in practice, the overall system behavior features elements from both the discrete and continuous domain. While a hybrid system model provides a natural abstraction for such behaviors, the problem of controller synthesis for hybrid systems suffers from both theoretical and computational difficulties. The results presented in this dissertation, whether theoretical, computational, or experimental, can be viewed as part of an overall effort towards systematic design and algorithmic synthesis of feedback control policies to satisfy reachability specifications for hybrid system abstractions of control systems. The methods that we have developed vary in terms of the model of interaction between the discrete and continuous dynamics, as well as of the model of uncertainty in the system behavior, and are to a large degree motivated by the needs of varying application scenarios. In the following, we will provide a summary of our main results, along with a discussion of directions for future work.

6.1 Summary

The controller design and synthesis methods developed in the first part of this dissertation are concerned with deterministic switched nonlinear systems in which the discrete transitions are controlled. These methods can be viewed as translations of the abstract controller synthesis algorithms for general hybrid systems as proposed in Lygeros et al. (1999*b*) and Tomlin et al. (2000), which have been largely applied manually on a case by case basis, to systematic design procedures and compuational synthesis techniques for subclasses of hybrid system models found in practical applications. In particular, we considered switched system models with two different interpretations of the discrete transitions as either the temporal phases of a dynamic process, or the control choices available to a high level controller. In the case of the former, a hybrid system formalism is proposed for sequential transition systems, whose discrete transitions follow a pre-defined sequence and can be controlled either by automation, or by an external human operator. For this class of systems, a systematic procedure is presented for designing continuous control laws and discrete switching conditions to satisfy sequential reachability specifications, namely specifications consisting of a sequence of safety or reach-avoid objective. This procedure uses the method of Hamilton-Jacobi

reachability analysis, as developed by Mitchell et al. (2005) for nonlinear continuous time systems, to compute reachable sets which provide information on the satisfaction of a safety or reach-avoid objective under a given continuous feedback policy, as well as the domain on which it is satisfied. This information is then used to design a feedback policy, which consists of both a continuous and a discrete component, to achieve individual reachability objectives within the respective modes, while assuring compatibility across mode transitions. The design procedure is applied to the example of automated aerial refueling (AAR). Simulation studies show that the resulting control design satisfies the desired safety and target attainability objectives, and that there is a possibility for using reachable sets as a guidance tool for decision making by a human operator.

For switched systems in which the discrete modes represent high level control choices, we described methods for performing algorithmic controller synthesis to satisfy safety and reach-avoid objectives, within a sampled-data setting, and subject to bounded continuous disturbances. In particular, under a discretization of the continuous input space, computational reachability analysis is performed over successive sampling intervals (once again using the Hamilton-Jacobi method), to find the set of initial conditions for which a given reachability specification is feasible, assuming feedback selections of the discrete mode and continuous input, and worst-case disturbance behavior. From the results of this reachability analysis, a set-valued feedback law is derived over the time horizon of interest, represented as a finite collection of reachable sets, along with an algorithm for performing online control selections with respect to the computed control law. This synthesis technique is applied in an experimental setting to the problem of controlling a quadrotor to reach and then remain in a hover region over a moving ground target. The experimental results show that the reachability-based control laws exhibit strong robustness properties and are for the most part capable of ensuring the desired specifications, excepting occasional, brief violations due to underestimation in the disturbance bound. It is also discussed how the methodology can be extended to perform controller synthesis for sequential reachability problems, by appropriately incorporating the design procedure of chapter 2 to ensure compatibility across the different phases of the reachability specification. This combined design approach is applied to the experimental example for the two phases of reach and hover, as well as to the AAR example in a simulation study, for the phases of the refueling sequence.

The theoretical and computational tools developed in the second part of this dissertation are concerned with discrete time stochastic hybrid systems (DTSHS), as motivated by applications in which system models are derived from statistical analysis or assumptions. Based upon prior work by Amin et al. (2006), Abate et al. (2006), and Summers et al. (2011) on probabilistic reachability problems for DTSHS, we considered two extensions to account for different models of uncertainty. In the first extension, two-player stochastic game formulations of the probability reachability problem are analyzed, in terms of a model which we referred to as a discrete-time stochastic hybrid game (DTSHG). These formulations feature a control whose objective is to achieve either a probabilistic safety or reach-avoid objective, and an adversarial disturbance whose objective is assumed to be opposed to that of the control. Our analysis of these formulations generalizes the stochastic optimal control argument used by Amin et al. (2006), Abate et al. (2006), and Summers et al. (2011) for the single player case, while also adapting results from the literature on additive cost stochastic games (see for example Kumar and Shiau, 1981; Nowak, 1985; Gonzalez-Trejo et al.,

2002) to the multiplicative payoff structure of the safety and reach-avoid problems. For a feedback Stackelberg formulation, with an asymmetric information pattern favoring the disturbance, a dynamic programming algorithm is given for the computation of the max-min probability of satisfying either the safety or the reach-avoid specifications, as the Stackelberg value. Sufficient conditions of optimality are also derived for the synthesis of a Stackelberg solution for the control, as a deterministic Markov policy. The Stackelberg value is later shown to be the lower value of a symmetric feedback Nash game, whose equilibrium solutions are in general found within the class of randomized policies. Some results on infinite horizon reachability computation are also provided, as related to the approximation of infinite horizon value and the existence of infinite horizon optimal policies. The utility of this methodology is illustrated using a two aircraft collision avoidance example, in which the adversarial uncertainty is due to the unknown intent of an uncontrolled aircraft, and the stochastic uncertainty is due to wind effects. Probabilistic reachability computations are performed to determine a conflict resolution strategy which provides probabilistic assurances of safety.

The second extension involves the consideration of partial information in probabilistic reachability problems, as pertaining to application scenarios in which control decisions are to be made with respect to uncertainties in the state estimate or measurement. These uncertainties can arise from limited sensor placements, lack of sensor precision, or measurement noise. As compared with the large body of previous work on perfect information reachability problems, there have been relatively few studies on the issue of imperfect state information in the literature on hybrid system reachability. In this dissertation, we investigated partial information safety and reach-avoid problems within the context of a partially observable discrete time stochastic hybrid system (POdt-SHS), which formally accounts for the imperfections in state measurement through a probabilistic observation model. Our analysis shows that a sufficient statistic for the partial information safety or reach-avoid problems maintains inferred knowledge about both the current state of the hybrid system and the safety of past state evolution. The added information, as encoded in a binary random variable augmenting the hybrid state space, differentiates the partial information reachability problems from conventional additive cost partial information problems, such as the LQG problem. Through a sequence of transformations which reduces the original partial information reachability problems to perfect information terminal cost or additive cost problems on the space of information states, we apply the results of Bertsekas and Shreve (1978) to derive a dynamic programming algorithm. It is then shown in the case of a POMDP model, with discrete state, control, and observation spaces, the information state is a probability distribution over twice the number of discrete states as the original model, and hence finite dimensional. However, in the case of a stochastic hybrid system model with probability density descriptions over continuous state spaces, the space of information states is in general infinite dimensional. Thus, computational solutions would have to be found in particular instances with finite dimensional representations or approximations for the hybrid conditional density.

6.2 Future Work

While we have taken some important first steps, the transition of reachability-based controller design methods from research and development to implementation in practical applications will require resolving a number of theoretical and computational issues. While some of these issues were alluded to in the individual chapters, we will highlight here some of the main challenges, and offer some ideas for future research.

6.2.1 Approximation of Deterministic Reachable Sets

By using the Hamilton-Jacobi approach for reachable set computation within each mode of a switched system, the controller design methods described in Part I of this dissertation can accommodate system models with up to four or five continuous state dimensions. While this may be sufficient for abstractions of control systems with a small number continuous states, as in the aircraft conflict resolution and AAR example, or applications in which there is decoupling in the continuous dynamics, as in the quadrotor target tracking example, there may be practical scenarios in which one may not be able to capture the richness of continuous behavior using a low dimensional model. For such cases it is important to investigate alternative methods for the reachable set computation.

Due to the fact that reachable sets for general nonlinear systems can exhibit increasingly complex shapes as one allows for increasing degrees of freedom, it comes as no surprise that numerical techniques for approximating such sets in general features a trade-off between accuracy and computational efficiency. While the Hamilton-Jacobi method can provide highly accurate approximations of reachable sets for low dimensional systems, it is also limited by its exponential growth in computational complexity. Some proposed methods for approximation of reachable sets for nonlinear systems include Mitchell and Tomlin (2003), Stipanović et al. (2004), Hwang et al. (2005), and Mitchell (2011), with varying forms of reachable set representation and levels of conservatism in the approximation. It would be interesting to investigate the possible use of these methods for the reachability computations described in chapter 2 and 3, with appropriate modifications of the controller design techniques to account for approximation errors.

For hybrid system models whose continuous dynamics are linear, a number of alternative reachability analysis techniques are available for the computation of approximate reachable sets in continuous time, based upon representations such as polyhedra (Asarin et al., 2000*a*; Chutinan and Krogh, 2003), ellipsoids (Kurzhanski and Varaiya, 2000), and zonotopes (Girard, 2005). More recently, a method has been proposed by Kaynama and Oishi (2011) for approximate reachability analysis of linear time invariant systems using Schur-based decomposition. While not all of these methods explicitly consider dynamic game formulations of reachability problems, it should be noted that the design procedures of chapter 2 and the controller synthesis algorithms of chapter 3 only require reachable set computations under differential inclusions (with respect to the disturbance input). However, as noted by Mitchell (2007*b*), there are subtle differences between these methods in their abilities to handle reachable set computation under existentially quantified inputs or universally quantified inputs. Given that our design method employs both capture sets,

which are computed under universally quantified disturbance inputs, and unsafe sets, which are computed under existentially quantified disturbance inputs, one should be careful in the selection of an approximation method which allows for both types of reachability computations.

6.2.2 Approximation of Probabilistic Reachability Computations

In an analogous fashion as the deterministic case, the applicability of the controller design methods described in chapter 4 for DTSHG models depends on one's ability to approximate the max-min safety and reach-avoid probabilities. Using a piecewise constant approximation, as adapted from the approach described in Abate et al. (2007) for the single player case, we have been able to perform probabilistic reachability computations in low dimensional examples. However, this method suffers from a similar type of exponential growth in complexity as the Hamilton-Jacobi method, due to the choice of a uniform grid over the hybrid state space. Thus, a more computationally efficient approximation algorithm, with provable bounds on the approximation error will be needed for problems with large continuous state dimensions. A possible approach is to investigate extensions of methods that have been developed in the realm of approximate dynamic programming. For example, various approaches have been proposed for using adaptive gridding of the state space (Munos and Moore, 2002), or parameterized families of basis functions (Bertsekas and Tsitsiklis, 1996; de Farias and van Roy, 2003; Kveton et al., 2006) to approximate the optimal value function in deterministic or stochastic optimal control problems. The difficulty in applying these approaches, however, lies in finding suitable adaptive grids and basis functions which result in accurate and tractable computation algorithms for probabilistic reachability problems. One effort in this direction can be found in the work of Esmaeil Zadeh Soudjani and Abate (2011) which describes an adaptive mesh refinement method for the approximation of the optimal safety probability of a DTSHS in the single player case. Another approach that has been proposed recently by McEneaney (2011) interprets the dynamic programming operator for stochastic optimal control problems as an abstract semigroup operator on a max-plus algebra. Through this viewpoint, the value function for certain classes of problems, such as those with affine dynamics and additive quadratic cost functions, can be represented through a pointwise minimum of quadratic functions. It would be of interest to investigate whether this approach can be extended to the multiplicative indicator cost functions encountered in probabilistic reachability problems.

6.2.3 Computational Approaches to Partial Information Probabilistic Reachability Problems

While the results of chapter 5 provide us with important insights into the structure of partial information safety and reach-avoid problems, they also serve to highlight the challenge of optimal control when we do not have accurate measurements or estimates of the system state. In particular, even in the case that the system only features continuous dynamics (e.g. a linear Gaussian system), the information needed to perform optimal control is in general a conditional probability distribution over a hybrid state space, with two discrete states. Thus, finding computational

solutions to such problems requires further understanding of hybrid estimation and the representation of hybrid probability distributions. One possibility is to explore parameterized families of functions which provide accurate approximations to the conditional probability distribution. In the case that the parameterization is finite dimensional, then it may be possible to develop approximate dynamic programming algorithms to compute suboptimal control policies on the space of parameterizations, with bounds on the suboptimality. An alternative approach is to formulate methods for computing the optimal safety or reach-avoid probabilities with respect to particular choices of estimators. Performance comparisons can be then made across the different estimators to decide on a final design. It is important to note that for the special case in which the discrete mode is known, and the continuous state estimation error is bounded, one can potentially include the estimation error as part of the disturbance and hence address the problem within the framework of a DTSHG. To illustrate this, consider a POdtSHS described as follows.

$$\begin{aligned} q(k+1) &\sim \nu_q(\cdot|(q(k),x(k)),(\sigma(k),u(k))),\ q(k) \in Q, \\ x(k+1) &= f(q(k),x(k),u(k),w(k)),\ x(k) \in \mathbb{R}^n, \\ o(k) &= q(k),\ y(k) = x(k) + v(k), \end{aligned} \tag{6.1}$$

where $o(k)$ is the discrete observation, $y(k)$ is the continuous observation, $w(k)$ is the process noise, and $v(k)$ is the continuous state measurement or estimation error, with the probability distribution $P_v(dv)$ over $\mathbb{R}^n$. Now suppose that the probability distribution of the estimation error has compact support, namely $P_v(B) = 1$ for some compact Borel set $B \in \mathscr{B}(\mathbb{R}^n)$. We can rewrite the model in (6.1) as

$$\begin{aligned} o(k+1) &\sim \nu_q(\cdot|(o(k),y(k)-v(k)),(\sigma(k),u(k))),\ o(k) \in Q, \\ y(k+1) &= f(o(k),y(k)-v(k),u(k),w(k)) + v(k+1),\ y(k) \in \mathbb{R}^n, \end{aligned} \tag{6.2}$$

If one were to treat the observation error in a worst-case fashion, then (6.2) describes a DTSHG model with the disturbance $b(k) = [v(k)\ v(k+1)]^T \in B^2$, and the methodology in chapter 4 of this dissertation applies. However, it can be seen that if the set B is large with respect to the reachability specification of interest, the results can be conservative. Furthermore, in the case that the observation error does not feature bounded support, for example in the case of a Gaussian distribution, then a different analysis technique is required in order to quantify the safety or reach-avoid probability.

6.2.4 Consideration of Multi-Objective Problems

For many safety-critical control applications, the performance specifications consist of both constraint satisfaction and cost minimization objectives. The former objectives are often of primary importance in ensuring safe and correct system behavior, and have been studied in this dissertation in the form of safety and reach-avoid problems, with proper interpretation for reachability specifications as state constraints and input spaces as control constraints. The latter objectives, on the other hand, ensure that the controller does not consume more resources than it needs to satisfy the

constraints. For example, in an aircraft conflict resolution scenario, one would like to ideally design controllers which generate safe and fuel-efficient trajectories. In Lygeros et al. (1999*b*), an abstract design methodology is proposed for a general class of hybrid systems, whereby the different control objectives are considered in a sequence of design steps according to their order of importance, with the controller design from the more important objectives serving as constraints for the less important objectives. Within the context of the controller synthesis algorithms for sampled-data switched systems (chapter 3) and DTSHG models (chapter 4), this idea can be potentially implemented through an added layer of dynamic programming procedure, by using the results of the reachability calculations as input constraints. In the switched system case, these constraints can be derived from the set-valued control laws given in sections 3.4 and 3.5, while in the DTSHG case, these constraints can be derived from the sufficient conditions of optimality given in sections 4.4 and 4.6. Some possible issues to such an approach, however, include whether the value functions of the resulting constrained optimal control problem satisfy necessary properties for dynamic programming, and also whether computationally tractable algorithms can be formulated.

6.2.5 Accounting for Autonomous Switches in Deterministic Continuous Time Hybrid Systems

Hybrid systems with controlled switching, as studied in Part I of this dissertation, are suitable for application scenarios in which the dynamics of a physical process is well-approximated by a nonlinear vector field (e.g. the kinematics of an aircraft), and the switching behavior is introduced through the discrete set of control choices (e.g. flight maneuvers) at the higher levels of abstraction. However, there is a number of instances in which autonomous discrete transitions, as triggered by changes in the continuous state, provides a natural abstraction of system behavior. This includes systems featuring event-triggered finite state machine models in high level control, pre-designed switching laws between modes of operation, or sharp changes in continuous dynamics caused by idealized physical modeling (e.g. elastic impacts). Examples of such systems range from automotive engines (Balluchi et al., 2000), power electronics (Aimer et al., 2007), to bipedal walkers (Ames et al., 2009).

However, the consideration of autonomous switching for continuous time systems is also accompanied by a significant increase in the difficulty of analyzing system properties. Specifically, state dependent switching introduces the possibility for discontinuous vector fields, which can result in infinitely fast switching or chattering at the switching boundary. This is referred to as a Zeno behavior in the hybrid systems literature (Zhang et al., 2001; Ames et al., 2005). The analysis of such scenarios typically requires generalized solution concepts for continuous trajectories (Filippov, 1988; Ames et al., 2006). While it may be reasonable in certain cases to work with models which preclude this behavior, the discontinuities in the vector field would nonetheless violate the typical assumptions of Lipschitz continuity in the analysis of viscosity solutions to HJB or HJI equations (Evans and Souganidis, 1984; Bardi and Capuzzo-Dolcetta, 1997).

There are several possibilities for overcoming this difficulty. One direction is to consider modifications of the Hamilton-Jacobi method for reachable set computation, whereby autonomous

switching is handled through proper definition of boundary conditions (Mitchell, 2002). Another possibility is to explore the use of explicit reachable set computation techniques which propagate sets directly under differential flows, and have been applied to examples with autonomous switching (see for example Asarin et al., 2000*a*; Chutinan and Krogh, 2003; Botchkarev and Tripakis, 2000; Girard and Le Guernic, 2008). As discussed previously, the controller design methods in Part I can accommodate alternative reachability analysis techniques as long as computations can be performed under both existentially and universally quantified inputs. Finally, it appears promising to apply existing methods based upon viability theory (Cardaliaguet et al., 1999; Aubin et al., 2002; Saint-Pierre, 2002; Gao et al., 2007). These methods have the advantage of being able to handle nonlinear dynamics, differential games, and discontinuous vector fields. Some disadvantages, as compared with the Hamilton-Jacobi approach, include the loss of value function information outside reachable sets, and the difficulty of assuring subgrid accuracy due to relaxation of the continuity assumption. The former is not a significant impediment to our controller design procedures, as we only require representations of sets, rather than value functions. The latter issue will require further investigation, as checking set-memberships for a given continuous state away from grid nodes may require numerical interpolation.

6.2.6 Reducing Conservativeness of Max-min Solutions for DTSHG

In our discussion of stochastic game formulations of the probabilistic reachability problems, we primarily assumed an asymmetric information pattern which favors the adversary. While this allows us to provide robust performance guarantees with respect to the worst-case adversary behavior, the resulting control policy can be also somewhat conservative, as it assumes that choices of control are revealed to the adversary at each discrete time instant. However, as we discussed in section 4.5.2, if one were to consider a symmetric zero-sum game formulation, the existence of an equilibrium solution would often require randomized policies, as opposed to the deterministic policies of an asymmetric formulation. This is in stark contrast with continuous time differential games, in which the saddle-point condition can be satified by a large class of nonlinear systems, whose dynamics are affine in the inputs (i.e. feedback linearizable systems). One intuitive explanation is that if one were to consider the discrete time system as a sampled continuous time system, then the choices of inputs on each sampling interval can be interpreted in terms of an open-loop rather than a feedback game. Thus, for discrete time models that are derived from continuous time ones, it would be interesting to investigate conditions under which the gap between the upper and lower values, in this case corresponding to the Stackelberg values of two asymmetric games, would become smaller as the time discretization is reduced.

Another possible direction is to explicitly consider the possible use of randomized policies. While they are not often found within the classical control applications, such type of control policies are of the norm in many modern communication protocols. In particular, the medium access control scheme of the IEEE 802.11 standard for wireless networks requires a station to select a random backoff time if it finds the channel to be busy when trying to transmit data, resulting in the well-known Markov chain model for the 802.11 family of protocols (Bianchi, 2000). When viewed from a multi-player game perspective, the reason for this randomization becomes some-

what intuitive. Namely, as each station with a packet to transmit is trying to gain access to a common medium (i.e. the channel) without knowledge of whether other stations might be transmitting during the same time slot, selection of randomized transmission times is one way to minimize the possibility of collision. However, it should also be noted that this very choice of randomization is one of the reasons that the performance of a wireless communication network is often much less predictable as compared with a traditional wired network. Thus, in deciding between deterministic and randomized policies for a given application, one should first consider the practical implications of a randomized approach. If such an approach is found to be reasonable, then the next step becomes addressing the computational issues. For cases in which the input space is continuous, the randomized policy space becomes infinite dimensional, namely it is the set of probability distributions over the input space. The problem then involves selecting a finite dimensional parameterization of a subclass of the randomized policies, and finding computationally tractable algorithms for carrying out the dynamic programming calculations with respect to the choice of parameterization.

6.2.7 Expanding the Class of Permissible Specifications

Our work in this dissertation has considered in some detail two types of reachability specifications. The first class is specifications with safety or invariance objectives, and the second class is those with target attainability objectives subject to a safety constraint. While they encompass a large range of atomic performance specifications encountered in practice, the complete design specifications of real-world control system often feature a combination of state-based objectives with temporal-based objectives. The sequential reachability specification considered in chapter 2 provides a simple example of this type of specifications. Namely, not only do we want to satisfy individual safety or reach-avoid specifications, we would also like them to be satisfied in a certain temporal order. Thus, the problem becomes one of composition between controllers satisfying individual safety or reach-avoid objectives. One of the immediate extensions would be to investigate the stochastic counterpart to this design approach for DTSHG models, in particular, whether nested dynamic programming algorithms can be carried out to compute the max-min probability of satisfying a sequence of safety and reach-avoid objectives.

Over the longer term, it would be interesting to explore whether sequential reachability specifications can be expanded to accommodate a richer class of temporal objectives, such as handled by discrete state model checking languages, including linear temporal logic (LTL), computation tree logic (CTL), and probabilistic computation tree logic (PCTL). The primitives of these languages, such as "always ϕ_1," "eventually ϕ_2," and "ϕ_1 until ϕ_2," where ϕ_1 and ϕ_2 are logic statements, have interpretations in terms of the reachability specifications discussed in this dissertation. Speaking somewhat informally, suppose that ϕ_1 is "remain in a safe set W" and ϕ_2 is "reach a target set R," then the statements given previously correspond to safety, terminal reachability, and reach-avoid, respectively. The power of these specification languages, however, lies in the combination of these primitives, along with logic operators to produce complex specifications such as "reach R_1 if a control command σ_1 is received after visiting R_2 or R_3, while always avoiding A_1 and A_2." For the interested reader, a comprehensive overview of LTL and CTL can be found in the survey

by Emerson (1990), while a detailed exposition on PCTL can be found in Hansson and Jonsson (1994).

Efforts to extend synthesis algorithms for temporal logic specifications to systems with continuous dynamics include the work by Belta et al. (2005); Tabuada and Pappas (2006); Kloetzer and Belta (2008); Fainekos et al. (2009); Kress-Gazit et al. (2009). These methods typically proceed by discretizing the state space into a finite number of partitions, and designing continuous controllers to satisfy objectives of staying within a partition or reach another partition in finite time. From the point of view of high level control, the partitions become the states of a discrete abstraction, with the continuous controllers implementing the state transitions in the discrete abstraction. The solution to the continuous synthesis problem can be then obtained from the result of a discrete synthesis algorithm. However, due to the difficulties of constructing continuous controllers implementing the required transition behaviors, applications of these methods have been mostly restricted to systems with affine continuous dynamics.

Combining the insights from these previous works with our experiences in addressing the sequential reachability specification, it would appear that the problem of synthesizing controllers to satisfy state-based objectives in conjunction with temporal-based objectives is inherently a hybrid control problem. Namely a discrete structure is induced by the temporal objectives over the set of atomic state-based reachability objectives. To be somewhat more concrete, in the case of the sequential reachability problem, the discrete structure is given by a sequence of transition states and stationary states, while the atomic reachability objectives are given by the reach-avoid objectives within the transition states and the invariance objectives within the stationary states. This then suggests a two staged approach to the synthesis problem, whereby the discrete structure is inferred from a given specification during the first stage, and the atomic controllers are constructed during the second stage, with proper considerations for composition between the atomic controllers. With improvements in the computational efficiency of reachability analysis, it is the hope of the author that such an approach would provide an avenue for addressing a range of interesting controller design problems arising in practical applications.

Bibliography

Abate, A., Amin, S., Prandini, M., Lygeros, J. & Sastry, S. (2006). Probabilistic reachability and safe sets computation for discrete time stochastic hybrid systems. In *Proceedings of the 45th IEEE Conference on Decision and Control*. San Diego, CA. pp. 258–263.

Abate, A., Amin, S., Prandini, M., Lygeros, J. & Sastry, S. (2007). Computational approaches to reachability analysis of stochastic hybrid systems. In *Hybrid Systems: Computation and Control* (A. Bemporad, A. Bicchi and G. Buttazzo, Eds.). Vol. 4416 of *Lecture Notes in Computer Science*. pp. 4–17. Springer Berlin / Heidelberg.

Abate, A., Katoen, J. P., Lygeros, J. & Prandini, M. (2010). Approximate model checking of stochastic hybrid systems. *European Journal of Control 16*(6), 624–641.

Abate, A., Prandini, M., Lygeros, J. & Sastry, S. (2008). Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica 44*(11), 2724–2734.

Ackerson, G. & Fu, K. (1970). On state estimation in switching environments. *IEEE Transactions on Automatic Control 15*(1), 10–17.

Aimer, S., Fujioka, H., Jonsson, U., Kao, C.-Y., Patino, D., Riedinger, P., Geyer, T., Beccuti, A., Papafotiou, G., Morari, M., Wernrud, A. & Rantzer, A. (2007). Hybrid control techniques for switched-mode DC-DC converters Part I: The step-down topology. In *Proceedings of the 2007 American Control Conference*. New York City, NY. pp. 5450–5457.

Alessandri, A. & Coletta, P. (2001). Design of Luenberger observers for a class of hybrid linear systems. In *Hybrid Systems: Computation and Control* (M. di Benedetto and A. Sangiovanni-Vincentelli, Eds.). Vol. 2034 of *Lecture Notes in Computer Science*. pp. 7–18. Springer Berlin / Heidelberg.

Altman, E. & Gaitsgory, V. (1997). Asymptotic optimization of a nonlinear hybrid system governed by a Markov decision process. *SIAM Journal on Control and Optimization 35*(6), 2070–2085.

Alur, R. & Dill, D. L. (1994). A theory of timed automata. *Theoretical Computer Science 126*(2), 183–235.

Alur, R., Courcoubetis, C. & Yannakakis, M. (1995). Distinguishing tests for nondeterministic and probabilistic machines. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*. ACM. Las Vegas, Nevada. pp. 363–372.

Alur, R., Courcoubetis, C., Henzinger, T. & Ho, P.-H. (1993). Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid Systems* (R. Grossman, A. Nerode, A. Ravn and H. Rischel, Eds.). Vol. 736 of *Lecture Notes in Computer Science*. pp. 209–229. Springer Berlin / Heidelberg.

Alur, R., Dang, T., Esposito, J., Fierro, R., Hur, Y., Ivančić, F., Kumar, V., Lee, I., Mishra, P., Pappas, G. & Sokolsky, O. (2001). Hierarchical hybrid modeling of embedded systems. In *Embedded Software* (T. Henzinger and C. Kirsch, Eds.). Vol. 2211 of *Lecture Notes in Computer Science*. pp. 14–31. Springer Berlin / Heidelberg.

Alur, R., Henzinger, T., Lafferriere, G. & Pappas, G. (2000). Discrete abstractions of hybrid systems. *Proceedings of the IEEE 88*(7), 971–984.

Ames, A., Abate, A. & Sastry, S. (2005). Sufficient conditions for the existence of Zeno behavior. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*. Seville, Spain. pp. 696–701.

Ames, A., Sinnet, R. & Wendel, E. (2009). Three-dimensional kneed bipedal walking: A hybrid geometric approach. In *Hybrid Systems: Computation and Control* (R. Majumdar and P. Tabuada, Eds.). Vol. 5469 of *Lecture Notes in Computer Science*. pp. 16–30. Springer Berlin / Heidelberg.

Ames, A., Zheng, H., Gregg, R. & Sastry, S. (2006). Is there life after Zeno? taking executions past the breaking (Zeno) point. In *Proceedings of the 2006 American Control Conference*. Minneapolis, MN. pp. 2652–2657.

Amin, S., Abate, A., Prandini, M., Lygeros, J. & Sastry, S. (2006). Reachability analysis for controlled discrete time stochastic hybrid systems. In *Hybrid Systems: Computation and Control* (J. Hespanha and A. Tiwari, Eds.). Vol. 3927 of *Lecture Notes in Computer Science*. pp. 49–63. Springer Berlin / Heidelberg.

Amin, S., Cárdenas, A. & Sastry, S. (2009). Safe and secure networked control systems under denial-of-service attacks. In *Hybrid Systems: Computation and Control* (R. Majumdar and P. Tabuada, Eds.). Vol. 5469 of *Lecture Notes in Computer Science*. pp. 31–45. Springer Berlin / Heidelberg.

Antsaklis, P., Stiver, J. & Lemmon, M. (1993). Hybrid system modeling and autonomous control systems. In *Hybrid Systems* (R. Grossman, A. Nerode, A. Ravn and H. Rischel, Eds.). Vol. 736 of *Lecture Notes in Computer Science*. pp. 366–392. Springer Berlin / Heidelberg.

Asarin, E., Bournez, O., Dang, T. & Maler, O. (2000*a*). Approximate reachability analysis of piecewise-linear dynamical systems. In *Hybrid Systems: Computation and Control* (N. Lynch

and B. Krogh, Eds.). Vol. 1790 of *Lecture Notes in Computer Science*. pp. 20–31. Springer Berlin / Heidelberg.

Asarin, E., Bournez, O., Dang, T., Maler, O. & Pnueli, A. (2000*b*). Effective synthesis of switching controllers for linear systems. *Proceedings of the IEEE 88*(7), 1011–1025.

Asarin, E., Maler, O. & Pnueli, A. (1995). Symbolic controller synthesis for discrete and timed systems. In *Hybrid Systems II* (P. Antsaklis, W. Kohn, A. Nerode and S. Sastry, Eds.). Vol. 999 of *Lecture Notes in Computer Science*. pp. 1–20. Springer Berlin / Heidelberg.

Aubin, J.-P., Lygeros, J., Quincampoix, M., Sastry, S. & Seube, N. (2002). Impulse differential inclusions: a viability approach to hybrid systems. *IEEE Transactions on Automatic Control 47*(1), 2–20.

Ballin, M. G. & Erzberger, H. (1996). An analysis of landing rates and separations at dallas/ft. worth airport. Technical Memorandum TM-110397. NASA.

Balluchi, A., Benvenuti, L., di Benedetto, M. & Sangiovanni-Vincentelli, A. (2002). Design of observers for hybrid systems. In *Hybrid Systems: Computation and Control* (C. Tomlin and M. Greenstreet, Eds.). Vol. 2289 of *Lecture Notes in Computer Science*. pp. 59–80. Springer Berlin / Heidelberg.

Balluchi, A., Benvenuti, L., di Benedetto, M., Pinello, C. & Sangiovanni-Vincentelli, A. (2000). Automotive engine control and hybrid systems: Challenges and opportunities. *Proceedings of the IEEE 88*(7), 888–912.

Bar-Shalom, Y. & Li., X.-R. (1993). *Estimation and Tracking: Principles, Techniques, and Software*. Artech House. Boston, MA.

Bardi, M. & Capuzzo-Dolcetta, I. (1997). *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*. Birkhäuser Boston. Boston, MA.

Başar, T. & Olsder, G. (1999). *Dynamic Noncooperative Game Theory*. Society for Industrial Mathematics.

Bayen, A., Mitchell, I., Oishi, M. & Tomlin, C. (2007). Aircraft autolander safety analysis through optimal control-based reach set computation. *AIAA Journal of Guidance, Control, and Dynamics 30*(1), 68–77.

Bect, J., Phulpin, Y., Baili, H. & Fleury, G. (2006). On the Fokker-Planck equation for stochastic hybrid systems: Application to a wind turbine model. In *International Conference on Probabilistic Methods Applied to Power Systems*. pp. 1–6.

Belta, C., Isler, V. & Pappas, G. (2005). Discrete abstractions for robot motion planning and control in polygonal environments. *IEEE Transactions on Robotics 21*(5), 864–874.

Bemporad, A., Ferrari-Trecate, G. & Morari, M. (2000*a*). Observability and controllability of piecewise affine and hybrid systems. *IEEE Transactions on Automatic Control 45*(10), 1864–1876.

Bemporad, A., Torrisi, F. & Morari, M. (2000*b*). Optimization-based verification and stability characterization of piecewise affine and hybrid systems. In *Hybrid Systems: Computation and Control* (N. Lynch and B. Krogh, Eds.). Vol. 1790 of *Lecture Notes in Computer Science*. pp. 45–58. Springer Berlin / Heidelberg.

Bensoussan, A. & Menaldi, J. (2000). Stochastic hybrid control. *Journal of mathematical analysis and applications 249*(1), 261–288.

Bertrand, N., Genest, B. & Gimbert, H. (2009). Qualitative determinacy and decidability of stochastic games with signals. In *Proceedings of the 24th Annual IEEE Symposium on Logic In Computer Science*. pp. 319–328.

Bertsekas, D. P. & Shreve, S. E. (1978). *Stochastic Optimal Control: The Discrete Time Case*. Academic Press. New York, NY.

Bertsekas, D. P. & Tsitsiklis, J. (1996). *Neuro-Dynamic Programming*. Athena Scientific. Belmont, MA.

Bianchi, G. (2000). Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications 18*(3), 535–547.

Blom, H. & Bar-Shalom, Y. (1988). The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control 33*(8), 780–783.

Blom, H. & Bloem, E. A. (2007). Exact Bayesian and particle filtering of stochastic hybrid systems. *IEEE Transactions on Aerospace and Electronic Systems 43*(1), 55–70.

Blom, H. & Lygeros, J. (2005). HYBRIDGE final project report. Technical report. National Aerospace Laboratory NLR, Amsterdam, The Netherlands.

Blom, H., Bakker, G. J. & Krystul, J. (2009). Rare event estimation for a large-scale stochastic hybrid system with air traffic application. In *Rare Event Simulation Using Monte Carlo methods* (G. Rubino and B. Tuffin, Eds.). Chap. 9, pp. 193–214. John Wiley & Sons.

Blom, H., Obbink, B. K. & Bakker, G. J. (2007). Safety risk simulation of an airborne self separation concept of operation. In *7th AIAA Aviation Technology, Integration and Operations Conference*. Belfast, Northern Ireland.

Botchkarev, O. & Tripakis, S. (2000). Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In *Hybrid Systems: Computation and Control* (N. Lynch and B. Krogh, Eds.). Vol. 1790 of *Lecture Notes in Computer Science*. pp. 73–88. Springer Berlin / Heidelberg.

Bottasso, C., Leonello, D. & Savini, B. (2008). Path planning for autonomous vehicles by trajectory smoothing using motion primitives. *IEEE Transactions on Control Systems Technology 16*(6), 1152–1168.

Branicky, M., Borkar, V. & Mitter, S. (1998). A unified framework for hybrid control: model and optimal control theory. *IEEE Transactions on Automatic Control 43*(1), 31–45.

Breton, M., Alj, A. & Haurie, A. (1988). Sequential Stackelberg equilibria in two-person games. *Journal of Optimization Theory and Applications 59*(1), 71–97.

Brockett, R. (1993). Hybrid models for motion control systems. In *Essays on Control: Perspectives in the Theory and its Applications* (H. L. Trentelman and J. C. Willems, Eds.). pp. 29–53. Birkhäuser Boston.

Brown, L. D. & Purves, R. (1973). Measurable selections of extrema. *The Annals of Statistics 1*(5), 902–912.

Buell, G. & Leondes, C. (1973). Optimal aircraft go-around and flare maneuvers. *IEEE Transactions on Aerospace and Electronic Systems AES-9*(2), 280–289.

Bujorianu, M. (2004). Extended stochastic hybrid systems and their reachability problem. In *Hybrid Systems: Computation and Control* (R. Alur and G. Pappas, Eds.). Vol. 2993 of *Lecture Notes in Computer Science*. pp. 234–249. Springer Berlin / Heidelberg.

Bujorianu, M. & Lygeros, J. (2004). General stochastic hybrid systems: modelling and optimal control. In *Proceedings of the 43rd IEEE Conference on Decision and Control*. Vol. 2. Paradise Island, Bahamas. pp. 1872–1877.

Burridge, R. R., Rizzi, A. A. & Koditschek, D. E. (1999). Sequential composition of dynamically dexterous robot behaviors. *The International Journal of Robotics Research 18*(6), 534–555.

Caines, P. & Wang, S. (1989). Classical and logic based regulator design and its complexity for partially observed automata. In *Proceedings of the 28th IEEE Conference on Decision and Control*. Vol. 1. Tampa, FL. pp. 132–137.

Caines, P. & Wei, Y.-J. (1998). Hierarchical hybrid control systems: a lattice theoretic formulation. *IEEE Transactions on Automatic Control 43*(4), 501–508.

Caines, P., Greiner, R. & Wang, S. (1988). Dynamical logic observers for finite automata. In *Proceedings of the 27th IEEE Conference on Decision and Control*. Vol. 1. Austin, Texas. pp. 226–233.

Callier, F. M. & Desoer, C. A. (1991). *Linear Systems Theory*. Springer-Verlag. New York, NY.

Campo, L., Mookerjee, P. & Bar-Shalom, Y. (1991). State estimation for systems with sojourn-time-dependent Markov model switching. *IEEE Transactions on Automatic Control 36*(2), 238–243.

Cardaliaguet, P., Quincampoix, M. & Saint-Pierre, P. (1999). Set-valued numerical analysis for optimal control and differential games. In *Stochastic and Differential Games* (M. Bardi, T. E. S. Raghavan, T. Parthasarathy and T. Başar, Eds.). Vol. 4 of *Annals of the International Society of Dynamic Games*. pp. 177–247. Birkhäuser Boston.

Chatterjee, K., Doyen, L. & Henzinger, T. (2010). Qualitative analysis of partially-observable Markov decision processes. In *Mathematical Foundations of Computer Science* (P. Hliněný and A. Kucera, Eds.). Vol. 6281 of *Lecture Notes in Computer Science*. pp. 258–269. Springer Berlin / Heidelberg.

Chatterjee, K., Doyen, L., Henzinger, T. & Raskin, J.-F. (2006). Algorithms for omega-regular games with imperfect information. In *Computer Science Logic* (Z. Ésik, Ed.). Vol. 4207 of *Lecture Notes in Computer Science*. pp. 287–302. Springer Berlin / Heidelberg.

Chutinan, A. & Krogh, B. H. (2003). Computational techniques for hybrid system verification. *IEEE Transactions on Automatic Control 48*(1), 64–75.

Cieslak, R., Desclaux, C., Fawaz, A. & Varaiya, P. (1988). Supervisory control of discrete-event processes with partial observations. *IEEE Transactions on Automatic Control 33*(3), 249–260.

Collins, P. & van Schuppen, J. (2004). Observability of piecewise-affine hybrid systems. In *Hybrid Systems: Computation and Control* (R. Alur and G. Pappas, Eds.). Vol. 2993 of *Lecture Notes in Computer Science*. pp. 265–279. Springer Berlin / Heidelberg.

Costa, E. F. & do Val, J. B. R. (2003). On the observability and detectability of continuous-time Markov jump linear systems. In *Proceedings of the 42nd IEEE Conference on Decision and Control*. Vol. 2. Maui, HI. pp. 1994–1999.

Crandall, M. G. & Lions, P.-L. (1983). Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society 277*(1), 1–42.

Crück, E. & Saint-Pierre, P. (2004). Nonlinear impulse target problems under state constraint: A numerical analysis based on viability theory. *Set-Valued Analysis 12*(4), 383–416.

Cummings, M. & Mitchell, P. (2008). Predicting controller capacity in supervisory control of multiple UAVs. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 38*(2), 451–460.

Davis, M. H. A. (1993). *Markov Models and Optimization*. Chapman & Hall/CRC Press. London.

de Alfaro, L., Henzinger, T. & Kupferman, O. (2007). Concurrent reachability games. *Theoretical Computer Science 386*(3), 188–217.

de Farias, D. P. & van Roy, B. (2003). The linear programming approach to approximate dynamic programming. *Operations Research 51*(6), 850–865.

De Wulf, M., Doyen, L. & Raskin, J.-F. (2006). A lattice theory for solving games of imperfect information. In *Hybrid Systems: Computation and Control* (J. Hespanha and A. Tiwari, Eds.). Vol. 3927 of *Lecture Notes in Computer Science*. pp. 153–168. Springer Berlin / Heidelberg.

Del Vecchio, D. (2009). Observer-based control of block-triangular discrete time hybrid automata on a partial order. *International Journal of Robust and Nonlinear Control 19*(14), 1581–1602.

Del Vecchio, D., Malisoff, M. & Verma, R. (2009). A separation principle for a class of hybrid automata on a partial order. In *Proceedings of the 2009 American Control Conference*. St. Louis, MO. pp. 3638–3643.

Ding, J. & Tomlin, C. (2010). Robust reach-avoid controller synthesis for switched nonlinear systems. In *Proceedings of the 49th IEEE Conference on Decision and Control*. Atlanta, GA. pp. 6481–6486.

Ding, J., Kamgarpour, M., Summers, S., Abate, A., Lygeros, J. & Tomlin, C. (2011*a*). A dynamic game framework for verification and control of stochastic hybrid systems. Technical Report UCB/EECS-2011-101. EECS Department, University of California, Berkeley.

Ding, J., Li, E., Huang, H. & Tomlin, C. (2011*b*). Reachability-based synthesis of feedback policies for motion planning under bounded disturbances. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*. Shanghai, China. pp. 2160–2165.

Ding, J., Sprinkle, J., Sastry, S. & Tomlin, C. (2008). Reachability calculations for automated aerial refueling. In *Proceedings of the 47th IEEE Conference on Decision and Control*. Cancun, Mexico. pp. 3706–3712.

Ding, J., Sprinkle, J., Tomlin, C., Sastry, S. & Paunicka, J. (2012). Reachability calculations for vehicle safety during manned/unmanned vehicle interaction. *AIAA Journal of Guidance, Control, and Dynamics 35*(1), 138–152.

Doucet, A., Logothetis, A. & Krishnamurthy, V. (2000). Stochastic sampling algorithms for state estimation of jump Markov linear systems. *IEEE Transactions on Automatic Control 45*(2), 188–202.

Emerson, E. A. (1990). Temporal and modal logic. In *Handbook of Theoretical Computer Science (Vol. B)* (J. van Leeuwen, Ed.). pp. 995–1072. MIT Press. Cambridge, MA.

Esmaeil Zadeh Soudjani, S. & Abate, A. (2011). Adaptive gridding for abstraction and verification of stochastic hybrid systems. In *Proceedings of the 8th International Conference on Quantitative Evaluation of Systems*. Aachen, Germany.

Evans, L. C. & Souganidis, P. E. (1984). Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations. *Indiana University Mathematics Journal 33*(5), 773–797.

Ezzine, J. & Haddad, A. H. (1988). On the controllability and observability of hybrid systems. In *Proceedings of the 1988 American Control Conference*. Atlanta, GA. pp. 41–46.

Fainekos, G., Girard, A., Kress-Gazit, H. & Pappas, G. (2009). Temporal logic motion planning for dynamic robots. *Automatica 45*(2), 343–352.

Fan, C.-H., Speyer, J. & Jaensch, C. (1994). Centralized and decentralized solutions of the linear-exponential-Gaussian problem. *IEEE Transactions on Automatic Control 39*(10), 1986–2003.

Fan, K. (1953). Minimax theorems. *Proceedings of the National Academy of Sciences 39*(1), 42–47.

Fierro, R., Das, A., Kumar, V. & Ostrowski, J. (2001). Hybrid control of formations of robots. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*. Vol. 1. Leuven, Belgium. pp. 157–162.

Filippov, A. F. (1988). *Differential Equations with Discontinuous Righthand Sides*. Kluwer Academic Publishers. Dordrecht, Netherlands.

Folland, G. B. (1999). *Real Analysis: Modern Techniques and Their Applications*. John Wiley & Sons. New York, NY.

Frazzoli, E., Dahleh, M. & Feron, E. (2000). Robust hybrid control for autonomous vehicle motion planning. In *Proceedings of the 39th IEEE Conference on Decision and Control*. Vol. 1. Sydney, Australia. pp. 821–826.

Frazzoli, E., Dahleh, M. & Feron, E. (2005). Maneuver-based motion planning for nonlinear systems with symmetries. *IEEE Transactions on Robotics 21*(6), 1077–1091.

Gao, Y., Lygeros, J. & Quincampoix, M. (2007). On the reachability problem for uncertain hybrid systems. *IEEE Transactions on Automatic Control 52*(9), 1572–1586.

Ghosh, R. & Tomlin, C. (2004). Symbolic reachable set computation of piecewise affine hybrid automata and its application to biological modelling: Delta-Notch protein signalling. *Systems Biology 1*(1), 170–183.

Girard, A. (2005). Reachability of uncertain linear systems using zonotopes. In *Hybrid Systems: Computation and Control* (M. Morari and L. Thiele, Eds.). Vol. 3414 of *Lecture Notes in Computer Science*. pp. 291–305. Springer Berlin / Heidelberg.

Girard, A. & Le Guernic, C. (2008). Zonotope/hyperplane intersection for hybrid systems reachability analysis. In *Hybrid Systems: Computation and Control* (M. Egerstedt and B. Mishra, Eds.). Vol. 4981 of *Lecture Notes in Computer Science*. pp. 215–228. Springer Berlin / Heidelberg.

Girard, A., Julius, A. & Pappas, G. (2008). Approximate simulation relations for hybrid systems. *Discrete Event Dynamic Systems 18*(2), 163–179.

Girard, A., Pola, G. & Tabuada, P. (2010). Approximately bisimilar symbolic models for incrementally stable switched systems. *IEEE Transactions on Automatic Control 55*(1), 116–126.

Glover, W. & Lygeros, J. (2004). A stochastic hybrid model for air traffic control simulation. In *Hybrid Systems: Computation and Control* (R. Alur and G. Pappas, Eds.). Vol. 2993 of *Lecture Notes in Computer Science*. pp. 372–386. Springer Berlin / Heidelberg.

Gollu, A. & Varaiya, P. (1989). Hybrid dynamical systems. In *Proceedings of the 28th IEEE Conference on Decision and Control*. Vol. 3. Tampa, FL. pp. 2708–2712.

Gonzalez-Trejo, J. I., Hernandez-Lerma, O. & Hoyos-Reyes, L. F. (2002). Minimax control of discrete-time stochastic systems. *SIAM Journal on Control and Optimization 41*(5), 1626–1659.

Gripon, V. & Serre, O. (2009). Qualitative concurrent stochastic games with imperfect information. In *Automata, Languages and Programming* (S. Albers, A. Marchetti-Spaccamela, Y. Matias, S. Nikoletseas and W. Thomas, Eds.). Vol. 5556 of *Lecture Notes in Computer Science*. pp. 200–211. Springer Berlin / Heidelberg.

Haghverdi, E., Tabuada, P. & Pappas, G. (2005). Bisimulation relations for dynamical, control, and hybrid systems. *Theoretical Computer Science 342*(2-3), 229–261.

Han, Z. & Krogh, B. (2006). Reachability analysis of nonlinear systems using trajectory piecewise linearized models. In *Proceedings of the 2006 American Control Conference*. Minneapolis, MN. pp. 1505–1510.

Hansson, H. & Jonsson, B. (1994). A logic for reasoning about time and reliability. *Formal Aspects of Computing 6*(5), 512–535.

Henzinger, T. (1996). The theory of hybrid automata. In *Proceedings of the 11th Annual IEEE Symposium on Logic In Computer Science*. pp. 278–292.

Henzinger, T. & Kopke, P. (1999). Discrete-time control for rectangular hybrid automata. *Theoretical Computer Science 221*(12), 369–392.

Henzinger, T., Ho, P.-H. & Wong-Toi, H. (1997). HYTECH: a model checker for hybrid systems. *International Journal on Software Tools for Technology Transfer (STTT) 1*(1), 110–122.

Henzinger, T., Kopke, P., Puri, A. & Varaiya, P. (1998). What's decidable about hybrid automata?,. *Journal of Computer and System Sciences 57*(1), 94–124.

Hespanha, J. (2004). Stochastic hybrid systems: Application to communication networks. In *Hybrid Systems: Computation and Control* (R. Alur and G. Pappas, Eds.). Vol. 2993 of *Lecture Notes in Computer Science*. pp. 47–56. Springer Berlin / Heidelberg.

Hofbaur, M. & Williams, B. (2002). Mode estimation of probabilistic hybrid systems. In *Hybrid Systems: Computation and Control* (C. Tomlin and M. Greenstreet, Eds.). Vol. 2289 of *Lecture Notes in Computer Science*. pp. 81–91. Springer Berlin / Heidelberg.

Hoffmann, G., Huang, H., Waslander, S. & Tomlin, C. (2007). Quadrotor helicopter flight dynamics and control: Theory and experiment. In *Proceedings of the AIAA Conference on Guidance, Navigation and Control*. Hilton Head, SC.

Hu, J., Lygeros, J. & Sastry, S. (2000). Towards a theory of stochastic hybrid systems. In *Hybrid Systems: Computation and Control* (N. Lynch and B. Krogh, Eds.). Vol. 1790 of *Lecture Notes in Computer Science*. pp. 160–173. Springer Berlin / Heidelberg.

Hu, J., Prandini, M. & Sastry, S. (2005). Aircraft conflict prediction in the presence of a spatially correlated wind field. *IEEE Transactions on Intelligent Transportation Systems 6*(3), 326–340.

Hu, J., Wu, W.-C. & Sastry, S. (2004). Modeling subtilin production in Bacillus subtilis using stochastic hybrid systems. In *Hybrid Systems: Computation and Control* (R. Alur and G. Pappas, Eds.). Vol. 2993 of *Lecture Notes in Computer Science*. pp. 163–166. Springer Berlin / Heidelberg.

Huang, H., Hoffmann, G., Waslander, S. & Tomlin, C. (2009). Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*. Kobe, Japan. pp. 3277–3282.

Hwang, I. & Seah, C. E. (2008). Intent-based probabilistic conflict detection for the next generation air transportation system. *Proceedings of the IEEE 96*(12), 2040–2059.

Hwang, I., Balakrishnan, H. & Tomlin, C. (2003). Observability criteria and estimator design for stochastic linear hybrid systems. In *Proceedings of the European Control Conference 2003*. Cambridge, UK.

Hwang, I., Stipanović, D. & Tomlin, C. (2005). Polytopic approximations of reachable sets applied to linear dynamic games and a class of nonlinear systems. In *Advances in Control, Communication Networks, and Transportation Systems* (E. H. Abed, Ed.). pp. 3–19. Systems and Control: Foundations and Applications. Birkhäuser Boston.

Isaacs, R. (1967). *Differential Games*. Wiley. New York, NY.

Jang, J. S. & Tomlin, C. (2005). Control strategies in multi-player pursuit and evasion game. In *Proceedings of the AIAA Conference on Guidance, Navigation and Control*. San Francisco, CA.

Jin, Z., Shima, T. & Schumacher, C. (2006). Scheduling and sequence reshuffle for autonomous aerial refueling of multiple UAVs. *Proceedings of the 2006 American Control Conference* pp. 2177–2182.

Kamgarpour, M., Ding, J., Summers, S., Abate, A., Lygeros, J. & Tomlin, C. (2011). Discrete time stochastic hybrid dynamic games: Verification & controller synthesis. In *Proceedings of the 50th IEEE Conference on Decision and Control, and the European Control Conference 2011*. Orlando, FL. pp. 6122–6127.

Kaynama, S. & Oishi, M. (2011). Complexity reduction through a schur-based decomposition for reachability analysis of linear time-invariant systems. *International Journal of Control 84*(1), 165–179.

Kloetzer, M. & Belta, C. (2006). Reachability analysis of multi-affine systems. In *Hybrid Systems: Computation and Control* (J. Hespanha and A. Tiwari, Eds.). Vol. 3927 of *Lecture Notes in Computer Science*. pp. 348–362. Springer Berlin / Heidelberg.

Kloetzer, M. & Belta, C. (2008). A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control 53*(1), 287–297.

Koo, T., Pappas, G. & Sastry, S. (2001). Mode switching synthesis for reachability specifications. In *Hybrid Systems: Computation and Control* (M. di Benedetto and A. Sangiovanni-Vincentelli, Eds.). Vol. 2034 of *Lecture Notes in Computer Science*. pp. 333–346. Springer Berlin / Heidelberg.

Koutsoukos, X. & Riley, D. (2006). Computational methods for reachability analysis of stochastic hybrid systems. In *Hybrid Systems: Computation and Control* (J. Hespanha and A. Tiwari, Eds.). Vol. 3927 of *Lecture Notes in Computer Science*. pp. 377–391. Springer Berlin / Heidelberg.

Koutsoukos, X., Kurien, J. & Zhao, F. (2003). Estimation of distributed hybrid systems using particle filtering methods. In *Hybrid Systems: Computation and Control* (O. Maler and A. Pnueli, Eds.). Vol. 2623 of *Lecture Notes in Computer Science*. pp. 298–313. Springer Berlin / Heidelberg.

Kress-Gazit, H., Fainekos, G. & Pappas, G. (2009). Temporal-logic-based reactive mission and motion planning. *IEEE Transactions on Robotics 25*(6), 1370–1381.

Kuchar, J. K. & Yang, L. C. (2000). A review of conflict detection and resolution modeling methods. *IEEE Transactions on Intelligent Transportation Systems 1*(4), 179–189.

Kumar, P. R. & Shiau, T. H. (1981). Existence of value and randomized strategies in zero-sum discrete-time stochastic dynamic games. *SIAM Journal on Control and Optimization 19*(5), 617–634.

Kumar, P. R. & van Schuppen, J. (1981). On the optimal control of stochastic systems with an exponential-of-integral performance index. *Journal of Mathematical Analysis and Applications 80*(2), 312–332.

Kumar, P. R. & Varaiya, P. (1986). *Stochastic Systems: Estimation, Identification, and Adaptive Control*. Prentice Hall. Englewood Gifts, NJ.

Kurzhanski, A. & Varaiya, P. (2000). Ellipsoidal techniques for reachability analysis. In *Hybrid Systems: Computation and Control* (N. Lynch and B. Krogh, Eds.). Vol. 1790 of *Lecture Notes in Computer Science*. pp. 202–214. Springer Berlin / Heidelberg.

Kushner, H. J. & Dupuis, P. (1992). *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer-Verlag. London, UK.

Kveton, B., Hauskrecht, M. & Guestrin, C. (2006). Solving factored MDPs with hybrid state and action variables. *Journal of Artificial Intelligence Research 27*, 153–201.

Lafferriere, G., Pappas, G. & Yovine, S. (1999). A new class of decidable hybrid systems. In *Hybrid Systems: Computation and Control* (F. Vaandrager and J. van Schuppen, Eds.). Vol. 1569 of *Lecture Notes in Computer Science*. pp. 137–151. Springer Berlin / Heidelberg.

Lainiotis, D. (1971). Optimal adaptive estimation: Structure and parameter adaption. *IEEE Transactions on Automatic Control 16*(2), 160–170.

Lam, T. M., Mulder, M. & van Paassen, M. M. (2008). Haptic feedback in uninhabited aerial vehicle teleoperation with time delay. *AIAA Journal of Guidance, Control, and Dynamics 31*(6), 1728–1739.

Larsen, K. G., Pettersson, P. & Yi, W. (1997). UPPAAL in a nutshell. *International Journal on Software Tools for Technology Transfer (STTT) 1*(1), 134–152.

Liberzon, D. & Morse, A. (1999). Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine 19*(5), 59–70.

Lin, F. & Wonham, W. (1988). On observability of discrete-event systems. *Information Sciences 44*(3), 173–198.

Lincoln, P. & Tiwari, A. (2004). Symbolic systems biology: Hybrid modeling and analysis of biological networks. In *Hybrid Systems: Computation and Control* (R. Alur and G. Pappas, Eds.). Vol. 2993 of *Lecture Notes in Computer Science*. pp. 147–165. Springer Berlin / Heidelberg.

Luenberger, D. (1971). An introduction to observers. *IEEE Transactions on Automatic Control 16*(6), 596–602.

Lusena, C., Goldsmith, J. & Mundhenk, M. (2001). Nonapproximability results for partially observable Markov decision processes. *Journal of Artificial Intelligence Research 14*, 83–103.

Lygeros, J. (1996). Hierarchical, Hybrid Control of Large Scale Systems. PhD thesis. University of California, Berkeley.

Lygeros, J., Godbole, D. & Sastry, S. (1998). Verified hybrid controllers for automated vehicles. *IEEE Transactions on Automatic Control 43*(4), 522–539.

Lygeros, J., Johansson, K., Sastry, S. & Egerstedt, M. (1999*a*). On the existence of executions of hybrid automata. In *Proceedings of the 38th IEEE Conference on Decision and Control*. Vol. 3. Phoenix, AZ. pp. 2249–2254.

Lygeros, J., Tomlin, C. & Sastry, S. (1999*b*). Controllers for reachability specifications for hybrid systems. *Automatica 35*(3), 349–370.

Magill, D. (1965). Optimal adaptive estimation of sampled stochastic processes. *IEEE Transactions on Automatic Control 10*(4), 434–439.

Maitra, A. & Parthasarathy, T. (1970). On stochastic games. *Journal of Optimization Theory and Applications 5*(4), 289–300.

Maitra, A. & Sudderth, W. (1998). Finitely additive stochastic games with Borel measurable payoffs. *International Journal of Game Theory 27*(2), 257–267.

Maler, O., Pnueli, A. & Sifakis, J. (1995). On the synthesis of discrete controllers for timed systems. In *STACS 95* (E. Mayr and C. Puech, Eds.). Vol. 900 of *Lecture Notes in Computer Science*. pp. 229–242. Springer Berlin / Heidelberg.

Maybeck, P. S. (1982). *Stochastic Models, Estimation and Control, Volume 2*. Academic Press. New York, NY.

McEneaney, W. M. (2011). Distributed dynamic programming for discrete-time stochastic control, and idempotent algorithms. *Automatica 47*(3), 443–451.

Merz, A. W. (1972). The game of two identical cars. *Journal of Optimization Theory and Applications 9*(5), 324–343.

Mitchell, I. (2002). Application of Level Set Methods to Control and Reachability Problems in Continuous and Hybrid Systems. PhD thesis. Stanford University.

Mitchell, I. (2007*a*). *A Toolbox of Level Set Methods*. `http://www.cs.ubc.ca/~mitchell/ToolboxLS/`.

Mitchell, I. (2007*b*). Comparing forward and backward reachability as tools for safety analysis. In *Hybrid Systems: Computation and Control* (A. Bemporad, A. Bicchi and G. Buttazzo, Eds.). Vol. 4416 of *Lecture Notes in Computer Science*. pp. 428–443. Springer Berlin / Heidelberg.

Mitchell, I. (2011). Scalable calculation of reach sets and tubes for nonlinear systems with terminal integrators: a mixed implicit explicit formulation. In *Proceedings of the 14th International Conference on Hybrid Systems: Computation and Control*. ACM. Chicago, IL. pp. 103–112.

Mitchell, I. & Tomlin, C. (2003). Overapproximating reachable sets by Hamilton-Jacobi projections. *Journal of Scientific Computing 19*(1), 323–346.

Mitchell, I., Bayen, A. & Tomlin, C. (2005). A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control 50*(7), 947–957.

Mohajerin Esfahani, P., Chatterjee, D. & Lygeros, J. (2011). On a problem of stochastic reach-avoid set characterization. In *Proceedings of the 50th IEEE Conference on Decision and Control, and the European Control Conference 2011*. Orlando, FL. pp. 7069–7074.

Munos, R. & Moore, A. (2002). Variable resolution discretization in optimal control. *Machine Learning 49*(2), 291–323.

Nalepka, J. P. & Hinchman, J. L. (2005). Automated aerial refueling: Extending the effectiveness of unmanned air vehicles. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*. San Francisco, CA.

Nash, J. (1951). Non-cooperative games. *Annals of Mathematics 54*(2), 286–295.

Nerode, A. & Kohn, W. (1993). Models for hybrid systems: Automata, topologies, controllability, observability. In *Hybrid Systems* (R. Grossman, A. Nerode, A. Ravn and H. Rischel, Eds.). Vol. 736 of *Lecture Notes in Computer Science*. pp. 317–356. Springer Berlin / Heidelberg.

Nowak, A. S. (1985). Universally measurable strategies in zero-sum stochastic games. *The Annals of Probability 13*(1), 269–287.

Oishi, M., Mitchell, I., Bayen, A., Tomlin, C. & Degani, A. (2002). Hybrid verification of an interface for an automatic landing. *Proceedings of the 41st IEEE Conference on Decision and Control 2*, 1607–1613.

Oishi, M., Mitchell, I., Tomlin, C. & Saint-Pierre, P. (2006). Computing viable sets and reachable sets to design feedback linearizing control laws under saturation. In *Proceedings of the 45th IEEE Conference on Decision and Control*. San Diego, CA. pp. 3801–3807.

Osher, S. & Fedkiw, R. (2002). *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag. New York, NY.

Ozveren, C. & Willsky, A. (1990). Observability of discrete event dynamic systems. *IEEE Transactions on Automatic Control 35*(7), 797–806.

Paielli, R. A. & Erzberger, H. (1997). Conflict probability estimation for free flight. *AIAA Journal of Guidance, Control and Dynamics 20*(3), 588–596.

Papadimitriou, C. H. & Tsitsiklis, J. N. (1987). The complexity of Markov decision processes. *Mathematics of Operations Research 12*(3), 441–450.

Petrov, A. & Zubov, A. (1996). On applicability of the interacting multiple-model approach to state estimation for systems with sojourn-time-dependent Markov model switching. *IEEE Transactions on Automatic Control 41*(1), 136–140.

Prajna, S., Jadbabaie, A. & Pappas, G. (2007). A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control 52*(8), 1415–1428.

Prandini, M. & Hu, J. (2006). A stochastic approximation method for reachability computations. In *Stochastic Hybrid Systems* (H. Blom and J. Lygeros, Eds.). Vol. 337 of *Lecture Notes in Control and Information Sciences*. pp. 107–139. Springer Berlin / Heidelberg.

Prandini, M., Hu, J., Lygeros, J. & Sastry, S. (2000). A probabilistic approach to aircraft conflict detection. *IEEE Transactions on Intelligent Transportation Systems 1*(4), 199–220.

Ramadge, P. J. (1986). Observability of discrete event systems. In *Proceedings of the 25th IEEE Conference on Decision and Control*. Vol. 25. Athens, Greece. pp. 1108–1112.

Reif, J. H. (1984). The complexity of two-player games of incomplete information. *Journal of Computer and System Sciences 29*(2), 274–301.

Rieder, U. (1991). Non-cooperative dynamic games with general utility functions. In *Stochastic Games and Related Topics* (T. Raghavan, T. S. Ferguson, T. Parthasarathy and O. J. Vrieze, Eds.). pp. 161–174. Kluwer Academic Publishers.

Ross, S., Pachter, M., Jacques, D., Kish, B. & Millman, D. (2006). Autonomous aerial refueling based on the tanker reference frame. *2006 IEEE Aerospace Conference* p. 22.

Rudin, W. (1976). *Principles of Mathematical Analysis, 3rd Edition*. McGraw-Hill. New York, NY.

Russell, S. J. & Norvig, P. (2002). *Artificial Intelligence: A Modern Approach, 2nd Edition*. Prentice Hall. Englewood Cliffs, NJ.

Saint-Pierre, P. (2002). Hybrid kernels and capture basins for impulse constrained systems. In *Hybrid Systems: Computation and Control* (C. Tomlin and M. Greenstreet, Eds.). Vol. 2289 of *Lecture Notes in Computer Science*. pp. 351–396. Springer Berlin / Heidelberg.

Sastry, S. (1999). *Nonlinear Systems: Analysis, Stability, and Control*. Springer-Verlag. New York, NY.

Sastry, S., Meyer, G., Tomlin, C., Lygeros, J., Godbole, D. & Pappas, G. (1995). Hybrid control in air traffic management systems. In *Proceedings of the 34th IEEE Conference on Decision and Control*. Vol. 2. New Orleans, LA. pp. 1478–1483.

Seah, C. E. & Hwang, I. (2009). Stochastic linear hybrid systems: Modeling, estimation, and application in air traffic control. *IEEE Transactions on Control Systems Technology 17*(3), 563–575.

Sethian, J. A. (1999). *Level Set Methods and Fast Marching Methods*. Cambridge University Press. New York, NY.

Shakernia, O., Pappas, G. & Sastry, S. (2001). Semi-decidable synthesis for triangular hybrid systems. In *Hybrid Systems: Computation and Control* (M. di Benedetto and A. Sangiovanni-Vincentelli, Eds.). Vol. 2034 of *Lecture Notes in Computer Science*. pp. 487–500. Springer Berlin / Heidelberg.

Shapley, L. S. (1953). Stochastic games. *Proceedings of the National Academy of Sciences 39*(10), 1095–1100.

Sion, M. (1958). Minimax theorems. *Pacific Journal of Mathematics 8*(1), 171–176.

Sontag, E. (1981). Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control 26*(2), 346–358.

Speyer, J., Deyst, J. & Jacobson, D. (1974). Optimization of stochastic linear systems with additive measurement and process noise using exponential performance criteria. *IEEE Transactions on Automatic Control 19*(4), 358–366.

Sprinkle, J., Ames, A., Eklund, J. M., Mitchell, I. & Sastry, S. (2005). Online safety calculations for glideslope recapture. *Innovations in Systems and Software Engineering 1*(2), 157–175.

Stipanović, D., Hwang, I. & Tomlin, C. (2004). Computation of an over-approximation of the backward reachable set using subsystem level set functions. In *Dynamics of Continuous, Discrete and Impulsive Systems*. Vol. 11 of *Series A: Mathematical Analysis*. pp. 399–411. Watam Press.

Summers, S. & Lygeros, J. (2010). Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem. *Automatica 46*(12), 1951–1961.

Summers, S., Kamgarpour, M., Lygeros, J. & Tomlin, C. (2011). A stochastic reach-avoid problem with random obstacles. In *Proceedings of the 14th international conference on Hybrid systems: computation and control*. ACM. Chicago, IL. pp. 251–260.

Sun, Z., Ge, S. S. & Lee, T. H. (2002). Controllability and reachability criteria for switched linear systems. *Automatica 38*(5), 775–786.

Szigeti, F. (1992). A differential-algebraic condition for controllability and observability of time varying linear systems. In *Proceedings of the 31st IEEE Conference on Decision and Control*. Vol. 4. Tucson, AZ. pp. 3088–3090.

Tabuada, P. (2008). An approximate simulation approach to symbolic control. *IEEE Transactions on Automatic Control 53*(6), 1406–1418.

Tabuada, P. & Pappas, G. (2006). Linear time logic control of discrete-time linear systems. *IEEE Transactions on Automatic Control 51*(12), 1862–1877.

Teo, R. (2005). Computing Danger Zones for Provably Safe Closely Spaced Parallel Approaches: Theory and Experiment. PhD thesis. Stanford University.

Teo, R. & Tomlin, C. (2003). Computing danger zones for provably safe closely spaced parallel approaches. *AIAA Journal of Guidance, Control, and Dynamics 26*(3), 434–443.

Thrun, S., Burgard, W. & Fox, D. (2005). *Probabilistic Robotics*. MIT Press. Cambridge, MA.

Tiwari, A. & Khanna, G. (2002). Series of abstractions for hybrid automata. In *Hybrid Systems: Computation and Control* (C. Tomlin and M. Greenstreet, Eds.). Vol. 2289 of *Lecture Notes in Computer Science*. pp. 425–438. Springer Berlin / Heidelberg.

Tiwari, A. & Khanna, G. (2004). Nonlinear systems: Approximating reach sets. In *Hybrid Systems: Computation and Control* (R. Alur and G. Pappas, Eds.). Vol. 2993 of *Lecture Notes in Computer Science*. pp. 171–174. Springer Berlin / Heidelberg.

Tomlin, C., Lygeros, J. & Sastry, S. (2000). A game theoretic approach to controller design for hybrid systems. *Proceedings of the IEEE 88*(7), 949–970.

Tomlin, C., Mitchell, I. & Ghosh, R. (2001). Safety verification of conflict resolution manoeuvres. *IEEE Transactions on Intelligent Transportation Systems 2*(2), 110–120.

Tomlin, C., Mitchell, I., Bayen, A. & Oishi, M. (2003). Computational techniques for the verification of hybrid systems. *Proceedings of the IEEE 91*(7), 986–1001.

Tomlin, C., Pappas, G. & Sastry, S. (2002). Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Transactions on Automatic Control 43*(4), 509–521.

Tugnait, J. K. (1982). Detection and estimation for abruptly changing systems. *Automatica 18*(5), 607–615.

Valasek, J., Kimmett, J., Hughes, D., Gunnam, K. & Junkin, J. L. (2002). Vision based sensor and navigation system for autonomous aerial refueling. In *AIAA's 1st Technical Conference and Workshop on Unmanned Aerospace Vehicles*. Portsmouth, VA.

Verma, R. & Del Vecchio, D. (2012). Safety control of hidden mode hybrid systems. *IEEE Transactions on Automatic Control 57*(1), 62–77.

Vidal, R., Chiuso, A., Soatto, S. & Sastry, S. (2003). Observability of linear hybrid systems. In *Hybrid Systems: Computation and Control* (O. Maler and A. Pnueli, Eds.). Vol. 2623 of *Lecture Notes in Computer Science*. pp. 526–539. Springer Berlin / Heidelberg.

von Neumann, J. & Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton University Press. Princeton, NJ.

Waydo, S., Hauser, J., Bailey, R., Klavins, E. & Murray, R. (2007). UAV as a reliable wingman: A flight demonstration. *IEEE Transactions on Control Systems Technology 15*(4), 680–688.

Whittle, P. (1981). Risk-sensitive linear/quadratic/Gaussian control. *Advances in Applied Probability 13*(4), 764–777.

Williamson, W. R., Glenn, G. J., Dang, V. T., Speyer, J. L., Stecko, S. M. & Takacs, J. M. (2009). Sensor fusion applied to autonomous aerial refueling. *AIAA Journal of Guidance, Control, and Dynamics 32*(1), 262–275.

Witsenhausen, H. (1966). A class of hybrid-state continuous-time dynamic systems. *IEEE Transactions on Automatic Control 11*(2), 161–167.

Wong-Toi, H. (1997). The synthesis of controllers for linear hybrid automata. In *Proceedings of the 36th IEEE Conference on Decision and Control*. Vol. 5. San Diego, CA. pp. 4607–4612.

Yang, L. C. & Kuchar, L. (1997). Prototype conflict alerting system for free flight. *AIAA Journal of Guidance, Control and Dynamics 20*(4), 768–773.

Yavrucuk, I., Unnikrishnan, S. & Prasad, J. (2009). Envelope protection for autonomous unmanned aerial vehicles. *AIAA Journal of Guidance, Control, and Dynamics 32*(1), 262–275.

Yovine, S. (1997). KRONOS: a verification tool for real-time systems. *International Journal on Software Tools for Technology Transfer (STTT) 1*(1), 123–133.

Zhang, J., Johansson, K. H., Lygeros, J. & Sastry, S. (2001). Zeno hybrid systems. *International Journal of Robust and Nonlinear Control 11*(2), 435–451.

Appendix A

Proof of Lemma 4.3

First we recall the notion of a simple function (see for example Folland, 1999).

Definition A.1. Let $(X, \mathscr{B}(X))$ be a borel space. A *simple function* on X is a finite linear combination, with complex coefficients, of characteristic functions of sets in $\mathscr{B}(X)$.

More concretely, a simple function is of the form $f = \sum_{k=1}^{K} z_k \mathbf{1}_{E_k}$, where $z_k \in \mathbb{C}$ and $E_k \in \mathscr{B}(X)$. Below is a result for approximation of measurable functions by simple functions, stated as Theorem 2.10(a) in Folland (1999).

Lemma A.1. *Let $(X, \mathscr{B}(X))$ be a borel space. If $f : X \to [0, \infty)$ is bounded and measurable, then there exists a sequence $\{\phi_n\}$ of simple functions with real coefficients such that $0 \leq \phi_1 \leq \phi_2 \leq \cdots \leq f$, and $\phi_n \to f$ uniformly on X.*

We will also need the following well-known result from real analysis (stated as Theorem 7.11 in Rudin (1976)).

Lemma A.2. *Let $\{f_n\}, n = 1, 2, \ldots$ and f be real-valued functions on a set E in a metric space X such that $f_n \to f$ uniformly E. Let x be a limit point of E, and suppose that*

$$\lim_{t \to x} f_n(t) = A_n$$

for $n = 1, 2, \ldots$. Then $\{A_n\}$ converges, and

$$\lim_{t \to x} f(t) = \lim_{n \to \infty} A_n$$

This result essentially allows an exchange of limits

$$\lim_{t \to x} \lim_{n \to \infty} f_n(t) = \lim_{n \to \infty} \lim_{t \to x} f_n(t)$$

when the convergence of f_n to f is uniform.

The proof now proceeds as follows.

Proof. With the observation that

$$\int f(y)t(dy|x) = \int f^+(y)t(dy|x) - \int f^-(y)t(dy|x)$$

where f^+ and f^- are the positive and negative parts of f, we can consider, without loss of generality, the case of $f \geq 0$.

Let x_0 be a limit point of X and $\{x_m\}_{m=1}^{\infty}$ be a sequence in X such that $x_m \to x_0$ as $m \to \infty$. For each $m \geq 0$, there exists a Borel-measurable function f_m on Y and a Borel subset B_m of Y such that $f = f_m$ on B_m and $t(B_m|x_m) = 1$ (Lemma 7.27 of Bertsekas and Shreve (1978)). Let $B = \bigcup_{m\geq 0} B_m$, then $t(B|x_m) = 1, \forall m \geq 0$. Define a function $\tilde{f} : B \to [0,\infty)$ by $\tilde{f}(x) = f_m(x)$, if $x \in B_m$. This definition is possible since for any m_1 and m_2 such that $x \in B_{m_1} \cap B_{m_2}$, we have

$$f_{m_1}(x) = f(x) = f_{m_2}(x)$$

Furthermore, $\tilde{f}$ is also Borel-measurable on B under the observation that for any Borel subset A of $[0,\infty)$

$$\tilde{f}^{-1}(A) = \bigcup_{m\geq 0} f_m^{-1}(A) \cap B_m.$$

By Lemma A.1, there exists a sequence of simple functions $\{\phi_n\}$ of the form $\phi_n = \sum_{k=1}^{K_n} z_k^n \mathbf{1}_{E_k^n}$, where $z_k^n \geq 0$ and $E_k^n \in \mathscr{B}(Y)$, such that $0 \leq \phi_1 \leq \phi_2 \leq \cdots \leq \tilde{f}$, and $\phi_n \to \tilde{f}$ uniformly on Y.

Define a function $g : X \to [0,\infty)$ as

$$g(x) = \int_B \tilde{f}(y)t(dy|x)$$

and functions $g_n : X \to [0,\infty)$, $n \in \mathbb{N}$ as

$$g_n(x) = \int_B \phi_n(y)t(dy|x)$$

Then by the Monotone Convergence Theorem (see for example Folland, 1999, Theorem 2.14), $g(x) = \lim_{n\to\infty} g_n(x)$, $\forall x \in X$. Furthermore, we claim that this convergence is uniform. Indeed, given the uniform convergence of ϕ_n to $\tilde{f}$, we have for every $\varepsilon > 0$ some $N \in \mathbb{N}$ such that

$$\tilde{f}(y) - \phi_n(y) < \varepsilon, \forall y \in Y, n \geq N$$

Thus, for any $x \in X$ and $n \geq N$, we have

$$g(x) - g_n(x) = \int_B (\tilde{f}(y) - \phi_n(y))t(dy|x) < \varepsilon$$

which completes the proof of the claim.

Now for each $m \geq 0$, $n \in \mathbb{N}$, the definition of Lebesgue integrals implies

$$g_n(x_m) = \int_B \phi_n(y)t(dy|x_m) = \sum_{k=1}^{K_n} z_k^n t(E_k^n|x_m)$$

By the continuity assumption on t, $t(E_k^n|x_m) \to t(E_k^n|x_0)$ as $m \to \infty$. Thus,

$$\lim_{m\to\infty} g_n(x_m) = \sum_{k=1}^{K_n} z_k^n t(E_k^n|x_0) = \int_B \phi_n(y) t(dy|x_0)$$

Applying Lemma A.2, we conclude that

$$\lim_{m\to\infty} \int_B \tilde{f}(y) t(dy|x_m) = \lim_{m\to\infty} g(x_m) = \lim_{n\to\infty} \int_B \phi_n(y) t(dy|x_0) = \int_B \tilde{f}(y) t(dy|x_0)$$

where the last equality follows by a repeated application of the Monotone Convergence Theorem. The statement of Lemma 4.3 now follows directly:

$$\begin{aligned} &\int_Y f(y) t(dy|x_m) = \int_B f(y) t(dy|x_m) = \int_B \tilde{f}(y) t(dy|x_m) \\ &\to \int_B \tilde{f}(y) t(dy|x_0) = \int_B f(y) t(dy|x_0) = \int_Y f(y) t(dy|x_0) \end{aligned}$$

as $m \to \infty$, which completes the proof. □

Appendix B

Proof of Proposition 4.7

Proof. From the proof of Lemma 4.5, we have that $r^N_{s_0}(R,W') = \mathscr{T}^N(\mathbf{1}_R)(s_0)$ is monotonically increasing for every $s_0 \in S$. Thus, by the definition of V_∞ in (4.47), it can be inferred that for each $N \geq 1$,

$$r^N_{s_0}(R,W') \leq V_\infty(s_0),\ \forall s_0 \in S.$$

By the monotonicity of the operator $\mathscr{T}$, it then follows that

$$r^{N+1}_{s_0}(R,W') \leq \mathscr{T}(V_\infty)(s_0),\ \forall s_0 \in S,\ N \in \mathbb{N}.$$

Taking the limit on the left hand side of this expression, we arrive at the inequality

$$V_\infty(s_0) \leq \mathscr{T}(V_\infty)(s_0),\ \forall s_0 \in S.$$

To show that the reverse inequality also holds, we define for notational convenience the functions $V_k : S \to [0,1]$ as $V_k := \mathscr{T}^k(\mathbf{1}_R)$, $k \geq 0$. Clearly, for every $s_0 \in S$,

$$\begin{aligned} V_\infty(s_0) &\geq \mathscr{T}^{N+1}(\mathbf{1}_R)(s_0) = \mathscr{T}(V_N)(s_0) \\ &\geq \inf_{b \in C_b} \mathbf{1}_R(s_0) + \mathbf{1}_{W' \setminus R}(s_0) H(s_0,a,b,V_N), \forall a \in C_a. \end{aligned} \tag{B.1}$$

By Proposition 4.1, there exists a Borel-measurable function $g^*_N : S \times C_a \to C_b$ which achieves the infimum in equation (B.1) for any fixed $(s_0,a) \in S \times C_a$. This then implies the inequality

$$V_\infty(s_0) \geq \mathbf{1}_R(s_0) + \mathbf{1}_{W' \setminus R}(s_0) H(s_0,a,g^*_N(s_0,a),V_N) \tag{B.2}$$

for every $s_0 \in S$, $a \in C_a$, and $N \geq 1$.

Given that the player II action space C_b is compact, the sequence $\{g^*_N(s_0,a)\}_{N=1}^\infty$ has a subsequence $\left\{g^*_{N_k}(s_0,a)\right\}_{k=1}^\infty$ which converges to some point $b^*_{(s_0,a)} \in C_b$ (see for example Rudin, 1976, Theorem 3.6). For any fixed $(s_0,a) \in S \times C_a$, we relabel the sequence $\left\{g^*_{N_k}(s_0,a)\right\}_{k=1}^\infty$ as $\left\{b^k_{(s_0,a)}\right\}_{k=1}^\infty$.

Now define a function $F^k : S \times C_a \times C_b \to [0,1]$ by

$$F^k(s_0,a,b) = \mathbf{1}_R(s_0) + \mathbf{1}_{W'\setminus R}(s_0) H(s_0,a,b,V_{N_k}).$$

Some useful properties of the operator H are given below:

- For any Borel-measurable functions $J, J' \in \mathscr{F}$ such that $J \leq J'$, $H(s,a,b,J) \leq H(s,a,b,J')$, $\forall s_0 \in S$, $a \in C_a$, $b \in C_b$.
- For any sequence of Borel-measurable functions $J_k \in \mathscr{F}$ such that $J_0 \leq J_k \leq J_{k+1}$ for all k and $\lim_{k\to\infty} J_k = J$, $\lim_{k\to\infty} H(s,a,b,J_k) = H(s,a,b,J)$, $\forall s_0 \in S$, $a \in C_a$, $b \in C_b$.

The first property can be directly inferred from the definition of H, while the second property follows by an application of the Monotone Convergence Theorem (see for example Folland, 1999, Theorem 2.14).

Using these properties and the fact that V_{N_k} is a sequence of monotonically increasing functions converging to V_∞, we have for every $s_0 \in S$, $a \in C_a$, $b \in C_b$ that

$$F^k(s_0,a,b) \leq F^{k+1}(s_0,a,b),\ k \geq 1 \tag{B.3}$$

$$\lim_{k\to\infty} F^k(s_0,a,b) = \mathbf{1}_R(s_0) + \mathbf{1}_{W'\setminus R}(s_0) H(s_0,a,b,V_\infty). \tag{B.4}$$

Consider a function $F : S \times C_a \times C_b \to [0,1]$ defined as

$$F(s_0,a,b) := \mathbf{1}_R(s_0) + \mathbf{1}_{W'\setminus R}(s_0) H(s_0,a,b,V_\infty).$$

Using (B.3) and (B.4), we will proceed to show that the following inequality holds:

$$\sup_{k\in\mathbb{N}} F^k(s_0,a,b^k_{(s_0,a)}) \geq F(s_0,a,b^*_{(s_0,a)}),\ \forall s_0 \in S, a \in C_a. \tag{B.5}$$

This combined with (B.2) would then imply

$$V_\infty(s_0) \geq \sup_{k\in\mathbb{N}} F^k(s_0,a,b^k_{(s_0,a)}) \geq F(s_0,a,b^*_{(s_0,a)}),$$

for every $s_0 \in S$ and $a \in C_a$, and hence

$$V_\infty(s_0) \geq \sup_{a\in C_a} \inf_{b\in C_b} F(s_0,a,b) = \mathscr{T}(V_\infty)(s_0),\ \forall s_0 \in S.$$

In order to show (B.5), we first observe that by (B.4),

$$\lim_{k\to\infty} F^k(s_0,a,b^*_{(s_0,a)}) = F(s_0,a,b^*_{(s_0,a)}),\ \forall s_0 \in S, a \in C_a.$$

Now fix $s_0 \in S$ and $a \in C_a$. For any $\varepsilon > 0$, it then follows that there exists some $N \in \mathbb{N}$ such that

$$F^N(s_0,a,b^*_{(s_0,a)}) \geq F(s_0,a,b^*_{(s_0,a)}) - \varepsilon.$$

By the assumptions placed on the DTSHG, $F^N(s_0, a, \cdot)$ is a continuous function on C_b, which implies

$$\lim_{k\to\infty} F^N(s_0, a, b^k_{(s_0,a)}) = F^N(s_0, a, b^*_{(s_0,a)}).$$

This in turn implies that there exists $K \in \mathbb{N}$ such that for every $k \geq K$,

$$F^N(s_0, a, b^k_{(s_0,a)}) \geq F^N(s_0, a, b^*_{(s_0,a)}) - \varepsilon.$$

Now we consider two cases. First, suppose $N \geq K$. Then

$$\begin{aligned} F^N(s_0, a, b^N_{(s_0,a)}) &\geq F^N(s_0, a, b^*_{(s_0,a)}) - \varepsilon \\ &\geq F(s_0, a, b^*_{(s_0,a)}) - 2\varepsilon \end{aligned}$$

Second, suppose $N < K$. Then by (B.3),

$$\begin{aligned} F^K(s_0, a, b^K_{(s_0,a)}) &\geq F^N(s_0, a, b^K_{(s_0,a)}) \\ &\geq F^N(s_0, a, b^*_{(s_0,a)}) - \varepsilon \\ &\geq F(s_0, a, b^*_{(s_0,a)}) - 2\varepsilon \end{aligned}$$

Since ε is arbitrary, (B.5) then follows. This completes the proof. □

Appendix C

Proof of Lemma 5.1

Proof. First, we note that given $\tilde{p} \in \mathscr{P}(\tilde{Q})$ and $a \in C_a$, the prediction equation (5.13) in section 5.3.2 becomes

$$\Psi(\tilde{p}, \sigma)(\tilde{q}') = \sum_{\tilde{q} \in \tilde{Q}} \tilde{p}_q(\tilde{q}'|\tilde{q}, \sigma)\tilde{p}(\tilde{q}),\ \tilde{q}' \in \tilde{Q}.$$

Furthermore, the stochastic kernels Φ_0 and Φ satisfying equations (5.11) and (5.12) in section 5.3.2 can be defined as

$$\Phi_0(\tilde{q}|\tilde{p}; o) = \Phi(\tilde{q}|\tilde{p}; o, \sigma) = \frac{\tilde{p}_o(o|\tilde{q})\tilde{p}(\tilde{q})}{\sum_{\tilde{q}' \in \tilde{Q}} \tilde{p}_o(o|\tilde{q}')\tilde{p}(\tilde{q}')},\ \tilde{q} \in \tilde{Q},$$

if $\sum_{\tilde{q}' \in \tilde{Q}} \tilde{p}_o(o|\tilde{q}')\tilde{p}(\tilde{q}') \neq 0$ and

$$\Phi_0(\tilde{q}|\tilde{p}; o) = \Phi(\tilde{q}|\tilde{p}; o, \sigma) = \bar{p}(\tilde{q}),\ \tilde{q} \in \tilde{Q},$$

where $\bar{p} \in \mathscr{P}(\tilde{Q})$ is arbitrary, if $\sum_{\tilde{q}' \in \tilde{Q}} \tilde{p}_o(o|\tilde{q}')\tilde{p}(\tilde{q}') = 0$. Now fix $p_0 \in \mathscr{P}(Q)$ and $\tilde{\pi}' \in \tilde{\Pi}'$. Define $\tilde{p}_k(\xi(p_0); i_k)$ recursively through the filtering equation (5.14) as

$$\begin{aligned}
&\tilde{p}_0(\xi(p_0); i_0)(\tilde{q}_0) = \Phi_0(\tilde{q}_0|\xi(p_0); o_0),\ \tilde{q}_0 \in \tilde{Q}, \\
&\tilde{p}_{k+1}(\xi(p_0); i_{k+1})(\tilde{q}_{k+1}) = \Phi(\tilde{q}_{k+1}|\Psi(\tilde{p}_k(\xi(p_0); i_k), \sigma_k); o_{k+1}, \sigma_k),\ \tilde{q}_{k+1} \in \tilde{Q},
\end{aligned}$$

Then by the definitions of Ψ, Φ_0 and Φ, it is sufficient to show that the following events

$$\begin{aligned}
E_0 &= \left\{ (\tilde{q}_0, o_0) \in \tilde{\Omega}_0 : \sum_{\tilde{q}_0' \in \tilde{Q}} \tilde{p}_o(o_0|\tilde{q}_0')\xi(p_0)(\tilde{q}_0') = 0 \right\}, \\
E_k &= \Big\{ (\tilde{q}_0, o_0, \sigma_0, ..., \tilde{q}_{k-1}, o_{k-1}, \sigma_{k-1}, \tilde{q}_k, o_k) \in \tilde{\Omega}_k : \\
&\qquad \sum_{\tilde{q}_k' \in \tilde{Q}} \sum_{\tilde{q}_{k-1}' \in \tilde{Q}} \tilde{p}_o(o_k|\tilde{q}_k')\tilde{p}_q(\tilde{q}_k'|\tilde{q}_{k-1}', \sigma_{k-1})\tilde{p}_{k-1}(\xi(p_0); i_{k-1})(\tilde{q}_{k-1}') = 0 \Big\},\ k \geq 1
\end{aligned}$$

have $\tilde{P}_k(\tilde{\pi}', \xi(p_0))$ measure zero for every k.

Indeed, for $k = 0$ and any set $\tilde{Q} \times \{o_0\} \subset E_0$, we have

$$\tilde{P}_0(\tilde{\pi}', \xi(p_0))(\tilde{Q} \times \{o_0\}) = \sum_{\tilde{q}_0 \in \tilde{Q}} \tilde{p}_o(o_0|\tilde{q}_0)\xi(p_0)(\tilde{q}_0) = 0,$$

from which it follows that $\tilde{P}_0(\tilde{\pi}', \xi(p_0))(E_0) = 0$.

For $k \geq 1$, we note that by Lemma 10.4 of Bertsekas and Shreve (1978), there exists a set $\tilde{I}_{k-1} \subset I_{k-1}$ with $\tilde{P}_{k-1}(\tilde{\pi}', \xi(p_0))$ measure one on which $\tilde{p}_{k-1}(\xi(p_0); i_{k-1})$ is the conditional probability distribution of $\tilde{q}_{k-1}$ given $\xi(p_0)$ and i_{k-1}. Thus, for any set $\tilde{Q}^{k+1} \times \{i_k\} \subset E_k$ such that $i_{k-1} \notin \tilde{I}_{k-1}$, $\tilde{P}_k(\tilde{\pi}', \xi(p_0))(\tilde{Q}^{k+1} \times \{i_k\}) = 0$. On the other hand, for any set $\tilde{Q}^{k+1} \times \{i_k\} \subset E_k$ such that $i_{k-1} \in \tilde{I}_{k-1}$,

$$\begin{aligned}
\tilde{P}_k(\tilde{\pi}', \xi(p_0))(\tilde{Q}^{k+1} \times \{i_k\}) = & \sum_{(\tilde{q}_0, \ldots, \tilde{q}_k) \in \tilde{Q}^{k+1}} \tilde{p}_o(o_k|\tilde{q}_k)\tilde{p}_q(\tilde{q}_k|\tilde{q}_{k-1}, \sigma_{k-1}) \\
& \times \tilde{\pi}'_{k-1}(\sigma_{k-1}|\xi(p_0); i_{k-1})\tilde{P}_{k-1}(\tilde{\pi}', \xi(p_0))(\tilde{q}_0, \ldots, \tilde{q}_{k-1}, i_{k-1}) \\
= & \sum_{(\tilde{q}_0, \ldots, \tilde{q}_{k-1}) \in \tilde{Q}^k} \sum_{\tilde{q}'_{k-1} \in \tilde{Q}} \sum_{\tilde{q}_k \in \tilde{Q}} \tilde{p}_o(o_k|\tilde{q}_k)\tilde{p}_q(\tilde{q}_k|\tilde{q}'_{k-1}, \sigma_{k-1}) \\
& \times \tilde{\pi}'_{k-1}(\sigma_{k-1}|\xi(p_0); i_{k-1})\tilde{p}_{k-1}(\xi(p_0); i_{k-1})(\tilde{q}'_{k-1}) \\
& \times \tilde{P}_{k-1}(\tilde{\pi}', \xi(p_0))(\tilde{q}_0, \ldots, \tilde{q}_{k-1}, i_{k-1}) = 0.
\end{aligned}$$

Hence, $\tilde{P}_k(\tilde{\pi}', \xi(p_0))(E_k) = 0$. The statement of the lemma then follows. □

Appendix D

Proof of Lemma 5.2

Proof. Fix any policy $\tilde{\pi}' \in \tilde{\Pi}'$. For the case of $k = 0$, consider a stochastic kernel $\Phi_0(\cdot|\xi(\bar{p}_0); z_0)$ defined by

$$\Phi_0(B_0|\xi(\bar{p}_0); z_0) = \frac{\int_{B_0} \tilde{p}_{z,0}(z_0|\tilde{s}_0)\xi(\bar{p}_0)(\tilde{s}_0)d\tilde{s}_0}{\int_{\tilde{S}} \tilde{p}_{z,0}(z_0|\tilde{s}'_0)\xi(\bar{p}_0)(\tilde{s}'_0)d\tilde{s}'_0},$$

for $B_0 \in \mathscr{B}(\tilde{S})$, if $\int_{\tilde{S}} \tilde{p}_{z,0}(z_0|\tilde{s}'_0)\xi(\bar{p}_0)(\tilde{s}'_0)d\tilde{s}'_0 \neq 0$ and

$$\Phi_0(B_0|\xi(\bar{p}_0); z_0) = \bar{p}(B_0),$$

for $B_0 \in \mathscr{B}(\tilde{S})$, if $\int_{\tilde{S}} \tilde{p}_{z,0}(z_0|\tilde{s}'_0)\xi(\bar{p}_0)(\tilde{s}'_0)d\tilde{s}'_0 = 0$, where $\bar{p} \in \mathscr{P}(\tilde{X})$ is arbitrary. By Proposition 7.29 of Bertsekas and Shreve (1978), the function $\lambda_0 : Z \to \mathbb{R}$ defined as

$$\lambda_0(z_0) = \int_{\tilde{S}} \tilde{p}_{z,0}(z_0|\tilde{s}'_0)\xi(\bar{p}_0)(\tilde{s}'_0)d\tilde{s}'_0$$

is Borel-measurable. Hence, the set

$$\bar{I}_0 := \left\{ z_0 \in Z : \int_{\tilde{S}} \tilde{p}_{z,0}(z_0|\tilde{s}'_0)\xi(\bar{p}_0)(\tilde{s}'_0)d\tilde{s}'_0 = 0 \right\}$$

is also Borel-measurable. It then follows that $\Phi_0(d\tilde{s}_0|\xi(\bar{p}_0); z_0)$ as defined above is a Borel-measurable stochastic kernel. Furthermore, it can be checked in a straightforward manner that Φ_0 satisfies equation (5.11) in section 5.3.2. By the filtering procedure in equation (5.14), we have

$$\tilde{p}_0(\xi(\bar{p}_0); z_0) = \Phi_0(d\tilde{s}'_0|\xi(\bar{p}_0); z_0).$$

Let $E_0 \subset \tilde{\Omega}_0$ be the set of events such that $\int_{\tilde{S}} \tilde{p}_{z,0}(z_0|\tilde{s}'_0)\xi(\bar{p}_0)(\tilde{s}'_0)d\tilde{s}'_0 = 0$. Clearly, for $(s_0, z_0) \notin E_0$, $\tilde{p}_0(\xi(\bar{p}_0); z_0)$ has the density $\tilde{p}_0^d(\cdot|\xi(\bar{p}_0); z_0)$ as given in the statement of the lemma. It remains to be shown that E_0 has $\tilde{P}_0(\tilde{\pi}', \xi(\bar{p}_0))$ measure zero. Indeed, by the observation that $E_0 = \tilde{S} \times \bar{I}_0$ and Fubini's theorem,

$$\begin{aligned}\tilde{P}_0(\tilde{\pi}', \xi(p_0))(E_0) &= \int_{\tilde{S}} \int_{\bar{I}_0} \tilde{p}_{z,0}(z_0|\tilde{s}'_0)\xi(\bar{p}_0)(\tilde{s}'_0)dz_0 d\tilde{s}'_0 \\ &= \int_{\bar{I}_0} \int_{\tilde{S}} \tilde{p}_{z,0}(z_0|\tilde{s}'_0)\xi(\bar{p}_0)(\tilde{s}'_0)d\tilde{s}'_0 dz_0 = 0.\end{aligned}$$

For the inductive step, we assume that for some $k \geq 1$, $\tilde{p}_{k-1}(\xi(\bar{p}_0); i_{k-1})$ has the probability density $\tilde{p}^d_{k-1}(\cdot|(\xi(\bar{p}_0); i_{k-1})$ for $\tilde{P}_{k-1}(\tilde{\pi}', \xi(\bar{p}_0))$ almost every i_{k-1}. We note that given $\tilde{p} \in \mathscr{P}(\tilde{S})$ and $a \in C_a$, the prediction equation (5.13) in section 5.3.2 becomes

$$\Psi(\tilde{p}, a)(\tilde{S}') = \int_{\tilde{S}} \left(\int_{\tilde{S}'} \tilde{p}_s(\tilde{s}'|\tilde{s}, a) d\tilde{s}' \right) \tilde{p}(d\tilde{s}),\ \tilde{S}' \in \mathscr{B}(\tilde{S}).$$

By an application of Fubini's theorem, $\Psi(\tilde{p}, a)$ has the density function

$$\tilde{s}' \to \int_{\tilde{S}} \tilde{p}_s(\tilde{s}'|\tilde{s}, a) \tilde{p}(d\tilde{s}).$$

Similarly as before, a stochastic kernel Φ satisfying equation (5.12) in section 5.3.2 can be defined as

$$\Phi(B|\tilde{p}; z, a) = \frac{\int_B \tilde{p}_Z(z|\tilde{s}, a) \tilde{p}(d\tilde{s})}{\int_{\tilde{S}} \tilde{p}_z(z|\tilde{s}', a) \tilde{p}(d\tilde{s}')},$$

for $B \in \mathscr{B}(\tilde{S})$, if $\int_{\tilde{S}} \tilde{p}_z(z|\tilde{s}', a) \tilde{p}(d\tilde{s}') \neq 0$ and

$$\Phi(B|\tilde{p}; z, a) = \bar{p}(B),$$

for $B \in \mathscr{B}(\tilde{S})$, if $\int_{\tilde{S}} \tilde{p}_z(z|\tilde{s}', a) \tilde{p}(d\tilde{s}') = 0$, where $\bar{p} \in \mathscr{P}(\tilde{S})$ is arbitrary. Define a set

$$C_k = \left\{ i_k \in I_k : \int_{\tilde{S}} \int_{\tilde{S}} \tilde{p}_z(z_k|\tilde{s}'_k, a_{k-1}) \tilde{p}_s(\tilde{s}'_k|\tilde{s}_{k-1}, a_{k-1}) \tilde{p}_{k-1}(\xi(\bar{p}_0); i_{k-1})(d\tilde{s}_{k-1}) d\tilde{s}'_k = 0 \right\}.$$

For $i_k \notin C_k$, we have by equation (5.14) in section 5.3.2 that for every $B \in \mathscr{B}(\tilde{S})$,

$$\begin{aligned} \tilde{p}_k(\xi(\bar{p}_0); i_k)(B) &= \Phi(B|\Psi(\tilde{p}_{k-1}(\xi(\bar{p}_0); i_{k-1}), a_{k-1}); z_k, a_{k-1}) \\ &= \frac{\int_B \tilde{p}_Z(z_k|\tilde{s}_k, a_{k-1}) \tilde{p}_{k|k-1}(\tilde{s}_k|\xi(\bar{p}_0); i_{k-1}, a_{k-1}) d\tilde{s}_k}{\int_{\tilde{S}} \tilde{p}_z(z_k|\tilde{s}'_k, a_{k-1}) \tilde{p}_{k|k-1}(\tilde{s}'_k|\xi(\bar{p}_0); i_{k-1}, a_{k-1}) d\tilde{s}'_k}, \end{aligned}$$

where

$$\tilde{p}_{k|k-1}(\tilde{s}_k|\xi(\bar{p}_0); i_{k-1}, a_{k-1}) = \int_{\tilde{S}} \tilde{p}_s(\tilde{s}_k|\tilde{s}_{k-1}, a_{k-1}) \tilde{p}_{k-1}(\xi(\bar{p}_0); i_{k-1})(d\tilde{s}_{k-1}).$$

By the induction hypothesis, there exists a set $\bar{I}_{k-1} \subset I_{k-1}$ with $\tilde{P}_{k-1}(\tilde{\pi}', \xi(\bar{p}_0))$ measure zero, away from which $\tilde{p}_{k-1}(\xi(\bar{p}_0); i_{k-1})$ has the probability density $\tilde{p}^d_{k-1}(\cdot|(\xi(\bar{p}_0); i_{k-1})$. It then follows that for $i_k \in \left(\bar{I}^C_{k-1} \times C_a \times Z\right) \cap C^C_k$, $\tilde{p}_k(\xi(\bar{p}_0); i_k)$ has the probability density $\tilde{p}^d_k(\cdot|(\xi(\bar{p}_0); i_k)$. With the observation that

$$\tilde{P}_k(\tilde{\pi}', \xi(\bar{p}_0))(S^k \times \bar{I}_{k-1} \times C_a \times S \times Z) = \tilde{P}_{k-1}(\tilde{\pi}', \xi(\bar{p}_0))(S^k \times \bar{I}_{k-1}) = 0,$$

it is sufficient to show that the set C_k has $\tilde{P}_k(\tilde{\pi}', \xi(\bar{p}_0))$ measure zero. For notational conveniences, we denote by $C_k(i_{k-1})$, $i_{k-1} \in I_{k-1}$ the i_{k-1}-section of C_k, namely

$$C_k(i_{k-1}) := \{(a_{k-1}, z_k) \in C_a \times Z : (i_{k-1}, a_{k-1}, z_k) \in C_k\}.$$

It can be checked that $C_k(i_{k-1})$ is a Borel subset of $C_a \times Z$ for every $i_{k-1} \in I_{k-1}$. By Lemma 10.4 of Bertsekas and Shreve (1978), there exists a subset $\tilde{I}_{k-1}$ of the information space I_{k-1} with $\tilde{P}_{k-1}(\tilde{\pi}', \xi(\bar{p}_0))$ measure one on which $\tilde{p}_{k-1}(\xi(\bar{p}_0); i_{k-1})$ is the conditional probability distribution of $\tilde{s}_{k-1}$ given $\xi(\bar{p}_0)$ and i_{k-1}. Using this fact and Fubini's theorem, we can deduce that

$$\begin{aligned}
\tilde{P}_k(\tilde{\pi}', \xi(\bar{p}_0))(\tilde{S}^{k+1} \times C_k) &= \int_{\tilde{S}^{k+1}} \int_{C_k} \tilde{p}_z(z_k|\tilde{s}_k) dz_k \tilde{p}_s(\tilde{s}_k|\tilde{s}_{k-1}, a_{k-1}) d\tilde{s}_k \\
&\quad \times \tilde{\pi}'_{k-1}(da_{k-1}|\xi(\bar{p}_0); i_{k-1}) d\tilde{P}_{k-1}(\tilde{\pi}', \xi(\bar{p}_0)) \\
&= \int_{\tilde{S}^k} \int_{I_{k-1}} \int_{\tilde{S}} \int_{C_k(i_{k-1})} \tilde{p}_z(z_k|\tilde{s}_k) dz_k \tilde{p}_s(\tilde{s}_k|\tilde{s}_{k-1}, a_{k-1}) \\
&\quad \times \tilde{\pi}'_{k-1}(da_{k-1}|\xi(\bar{p}_0); i_{k-1}) d\tilde{s}_k d\tilde{P}_{k-1}(\tilde{\pi}', \xi(\bar{p}_0)) \\
&= \int_{\tilde{S}^k} \int_{\tilde{I}_{k-1}} \int_{\tilde{S}} \int_{\tilde{S}} \int_{C_k(i_{k-1})} \tilde{p}_z(z_k|\tilde{s}'_k) dz_k \tilde{p}_s(\tilde{s}'_k|\tilde{s}_{k-1}, a_{k-1}) \\
&\quad \times \tilde{\pi}'_{k-1}(da_{k-1}|\xi(\bar{p}_0); i_{k-1}) d\tilde{s}'_k \tilde{p}_{k-1}(\xi(\bar{p}_0); i_{k-1})(d\tilde{s}_{k-1}) \\
&\quad \times d\tilde{P}_{k-1}(\tilde{\pi}', \xi(\bar{p}_0)) \\
&= \int_{\tilde{S}^k} \int_{\tilde{I}_{k-1}} \int_{C_k(i_{k-1})} \int_{\tilde{S}} \int_{\tilde{S}} \tilde{p}_z(z_k|\tilde{s}'_k) \tilde{p}_s(\tilde{s}'_k|\tilde{s}_{k-1}, a_{k-1}) \\
&\quad \times \tilde{p}_{k-1}(\xi(\bar{p}_0); i_{k-1})(d\tilde{s}_{k-1}) d\tilde{s}'_k dz_k \tilde{\pi}'_{k-1}(da_{k-1}|\xi(\bar{p}_0); i_{k-1}) \\
&\quad \times d\tilde{P}_{k-1}(\tilde{\pi}', \xi(\bar{p}_0)) = 0
\end{aligned}$$

The statement of the lemma then follows. □